JOURNAL OF INDUSTRIAL AND
MANAGEMENT OPTIMIZATION
Volume **13**, Number **4**, October **2017**

doi:10.3934/jimo.2017021

pp. **1841–1857**

INCREMENTAL GRADIENT-FREE METHOD FOR NONSMOOTH DISTRIBUTED OPTIMIZATION

Jueyou Li, Guoquan Li, Zhiyou Wu

School of Mathematical Sciences, Chongqing Normal University
Chongqing, 400047, China

Changzhi Wu, Xiangyu Wang

Australasian Joint Research Center for Building Information Modelling
School of Built Environment, Curtin University, Bentley, WA, 6102, Australia

Jae-Myung Lee and Kwang-Hyo Jung

Department of Naval Architecture and Ocean Engineering
Pusan National University, Busan, Korea

(Communicated by Paulo J. S. Silva)

Abstract. In this paper we consider the minimization of the sum of local convex component functions distributed over a multi-agent network. We first extend the Nesterov's random gradient-free method to the incremental setting. Then we propose the incremental gradient-free methods, including a cyclic order and a randomized order in the selection of component function. We provide the convergence and iteration complexity analysis of the proposed methods under some suitable stepsize rules. To illustrate our proposed methods, extensive numerical results on a distributed l_1-regression problem are presented. Compared with existing incremental subgradient-based methods, our methods only require the evaluation of the function values rather than subgradients, which may be preferred by practical engineers.

1. **Introduction.** In recent years, there is an increasing trend for minimizing the sum of a number of component functions that all share a common decision variable [17, 5, 18, 7]. Such problems are often termed distributed optimization [3, 18, 7], and they arise in many network applications, including in-network estimation, learning, signal processing, and resource allocation [21, 23, 24, 29, 31]. In these applications, there is no central coordinator that has access to all the information which traditional optimization approaches are required [27]. Thus, decentralized algorithms are needed to solve the problems.

2010 *Mathematics Subject Classification.* Primary: 47N10; Secondary: 49J52.

Key words and phrases. Incremental method, Gaussian smoothing, gradient-free method, convex optimization.

This research was partially supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIP) through GCRC-SOP (No. 2011-0030013), the Natural Science Foundation of China (11501070, 11401064, 11471062 and 61473326), by the Natural Science Foundation Projection of Chongqing (cstc2015jcyjA00011, cstc2013jjB00001 and cstc2013jcyjA00029), by the Chongqing Municipal Education Commission under Grant KJ1500301 and KJ1500302, and by the Chongqing Normal University Research Foundation 15XLB005.

There exist several useful techniques for solving optimization problems in a distributed manner. In terms of the update strategies, most of them can be classified as the consensus based approach [18, 7, 14] and the incremental based approach [25, 16, 17, 4, 13]. In the consensus based approach, the agents achieve the minimizer globally through sharing the information locally (the agent only shares information within its neighbors). In [18], Nedić developed a general framework for distributed optimization based on subgradient methods and the consensus strategy over a time-varying network. The authors in [7] discussed a constrained distributed optimization based on dual subgradient averaging. In [14], Li et. al extended the Nesterov's gradient-free method [19] to the distributed setting based on the consensus scheme. Later, the authors in [15] further developed a distributed proximal gradient method for solving a class of convex optimization with inequality constraints. In [26], Yuan et. al proposed a derivative-free distributed method for multi-agent optimization based on Nesterov's gradient-free method and push-sum strategy. In the incremental approach, the basic idea is to perform the (sub)gradient-based update incrementally, by sequentially taking steps along the (sub)gradients of the component functions, with intermediate adjustment of the variables after processing each component function [25, 17]. Hence, it can also work in a fully distributed manner and has been very successful in solving parameter estimation in networks of wireless sensors [21, 23], stochastic programming [8]. In terms of the strategies that component functions are selected, most of existing incremental based approach can be classified into as cyclic incremental methods [17, 5], equiprobable randomized incremental methods [17, 5], and Markov randomized incremental methods [12, 22]. In [17], the authors developed the cyclic incremental subgradient method and the equiprobable randomized incremental subgradient method for distribution optimization. Their contribution is to provide the explicit convergence rates of their proposed methods [16, 17]. Unlike the cyclic incremental method, the authors have extended equiprobable randomized incremental subgradient method to a more general case where the sequence in which component functions are selected and processed in a time homogeneous Markov chain [12] or a time nonhomogeneous Markov chain [22]. To minimize the sum of a number of composite component functions, the proximal point method is firstly incorporated in the incremental method [5].

Most of the methods mentioned above are based on the assumption that the subgradients of objective functions are available and easy to evaluate. However, it is well-known that there exist a large number of problems where the subgradient information is unavailable or costly to compute [6, 8, 19, 30]. On the other hand, for many practical engineers, derivative-free methods are always preferred since some substantial efforts of the subgradient computation require the knowledge of convex optimization and nonsmooth analysis. Thus, the development of derivative-free optimization schemes attracts many research interests. For the centralized optimization problems, derivative-free optimization schemes have a long history, which enjoy the desirable advantage of never requiring explicit subgradient calculations. For details we can refer to [1, 6, 8, 28, 19, 30]. However, there is still limitation on gradient-free methods available in the distributed setting. In this paper, based on the incremental approach, we develop gradient-free computational schemes for distributed optimization problems.

The structure of this paper is as follows. In Section 2, we formulate the problem under consideration. In Section 3, we propose a cyclic incremental gradient-free algorithm associated with a ring structure network and give the convergence analysis

of the algorithm. In Section 4, we propose a randomized incremental gradient-free algorithm over an arbitrary network and give the convergence results. Numerical experiments are presented in Section 5. Finally, some conclusions are given. Comparing with existing incremental-based methods, the incremental gradient-free algorithms over two types of networks (a cyclic network in Section 3 and an arbitrary network in Section 4) are considered in this paper. More importantly, our proposed methods are free of gradient, which may be preferred by practical engineers. Since only the values of cost functions are required, our method may suffer a factor of at most d^2 (d is the dimension of the problem variable) in iteration complexity over incremental subgradient-based methods in theory. However, our numerical simulations show that for some nonsmooth problems, our methods can even achieve better performance than that of subgradient-based methods under the same stepsize updating strategies.

2. Problem and Gaussian smoothing.

In this section, we formulate the problem of interest and describe the Gaussian smoothing technique that we will use.

2.1. Problem.

Consider a network with m agents, indexed by $i = 1, \ldots, m$. The network objective is to solve the following optimization problem:

$$\min \sum_{i=1}^{m} f_i(x) \text{ s.t. } x \in X, \tag{1}$$

where x is a global decision vector, $X \subseteq \mathbb{R}^d$ is a closed, convex set, and each $f_i(x) : X \to \mathbb{R}$ is a convex function but nonsmooth, only known by agent i.

In this paper, we are interested in the case when the cost function values of the problem (1) are only available. Our goal is to deal with the situation in which each agent i has only access to its private cost function value f_i ($i = 1, \ldots, m$), and all agents cooperatively minimizes the sum of convex objective functions of the agents over a multi-agent network.

For the simplicity of notation, we define

$$F(x) = \sum_{i=1}^{m} f_i(x), \ F^* = \min_{x \in X} F(x), \ X^* = \{x \in X : F(x) = F^*\}.$$

Throughout the paper, we assume that X^* is nonempty and F^* is finite. To proceed it further, we require the following assumption.

Assumption 1. *Each function f_i is L-Lipschitz with respect to l_2-norm $||\cdot||$, that is, there exists a constant $L > 0$ such that*

$$|f_i(x) - f_i(y)| \leq L||x - y||, \ \forall x, y \in X, \ i = 1, \ldots, m.$$

Note that Assumption 1 implies the boundedness of subgradient for the function f_i, i.e.,

$$||g_i(x)|| \leq L, \ \forall g_i(x) \in \partial f_i(x), x \in X, i = 1, \ldots, m,$$

where $\partial f_i(x)$ is the set of subgradients for the function $f_i(x)$ (see e.g., [10]).

2.2. **Gaussian smoothing.** In order to address difficulties associated with the nonsmooth objective function, we consider a smooth approximation of the objective function. It is well-known (see, e.g., Proposition 2.4 in [2]) that the convolution of two functions is at least as smooth as the smoother of the two original functions. In particular, let u be d-dimensional standard Gaussian random vector and $\nu > 0$ be the smoothing parameter, then a smooth approximation of a nonsmooth function f is defined by

$$f^\nu(x) = \frac{1}{(2\pi)^{\frac{d}{2}}} \int_{\mathbb{R}^d} f(x+\nu u) e^{-\frac{1}{2}||u||^2} du = \mathbb{E}_u[f(x+\nu u)].$$

In addition, we have other choices of the smoothing distribution. For example, the uniform distribution on the l_2-ball or l_∞-ball has been used in [8, 30, 9]. In what follows, we give some useful properties of $f^\nu(x)$.

Lemma 2.1. [19] *Assume that $f(x)$ is convex and L-Lipschitz on X, then, for any smoothing parameter $\nu > 0$, the following properties hold,*

(i): $f(x) \le f^\nu(x) \le f(x) + \nu\sqrt{d}L$;

(ii): *$f^\nu(x)$ is convex, continuously differentiable and the gradient $\nabla f^\nu(x)$ is given by*

$$\nabla f^\nu(x) = \frac{1}{(2\pi)^{\frac{d}{2}}} \int_{\mathbb{R}^d} \frac{f(x+\nu u) - f(x)}{\nu} u e^{-\frac{1}{2}||u||^2} du;$$

(iii): *for $\nu > 0$, define the following random gradient-free oracle:*

$$\tilde{g}(x,u) = \frac{f(x+\nu u) - f(x)}{\nu} u, \tag{2}$$

then $\mathbb{E}_u[\tilde{g}(x,u)] = \nabla f^\nu(x)$;

(iv): $\mathbb{E}_u[||\tilde{g}(x,u)||] \le dL$, $\mathbb{E}_u[||\tilde{g}(x,u)||^2] \le (d+4)^2 L^2$.

It can be seen from Lemma 2.1 (i) that f^ν is the *Gaussian approximation* of f. When compared with the nonsmooth function f, the smooth version f^ν is relatively well-behaved (see, Lemma 2.1 (ii)), moreover, there is a close relationship between its gradient $\nabla f^\nu(x)$ and the random gradient-free oracle $\tilde{g}(x,u)$ (see, Lemma 2.1 (iii)). Note, however, the bound $\mathbb{E}_u[||\tilde{g}(x,u)||]$ is scaled as d (see, Lemma 2.1 (iv)), which is an additional penalty induced by the use of the gradient-free oracle.

Lemma 2.2. [14] *Assume that $f(x)$ is convex and L-Lipschitz on X, then, for any smoothing parameter $\nu > 0$, we have*

$$f(x) - f(\bar{x}) \le \nabla f^\nu(x)^T (x - \bar{x}) + \nu\sqrt{d}L, \ \forall x, \bar{x} \in X.$$

3. **Incremental gradient-free method with cyclic order.** In this section, we first propose a cyclic incremental gradient-free algorithm for solving problem (1), then give the convergence results of the algorithm. Our method is based on the cyclic incremental subgradient method [4]. Our focus is on the case where the subgradient evaluations of the agents are unavailable or prohibitive.

We assume that all agents are connected in a directed network with ring structure [17], each agent merely exchange information with its direct neighbors. Formally, the cyclic incremental gradient-free (CIGF, for short) algorithm is described as follows:

Algorithm CIGF

Initialization: Given an initial point $x_0 \in X$, stepsizes $\{\alpha_{k+1}\}_{k\geq 0}$, smoothing parameters $\mu_i > 0, i = 1, \ldots, m$. Set $k = 0$.

Step 1: Update

$$z_{m,k} = z_{0,k+1} = x_k. \tag{3}$$

Step 2: For $i = 1, \ldots, m$, do

Step 2.1: Generate $u_{i,k+1}$ by Gaussian random vector generator and call the gradient-free oracle of the component function f_i for computing $\tilde{g}(z_{i-1,k+1}, u_{i,k+1})$ given by

$$\tilde{g}(z_{i-1,k+1}, u_{i,k+1}) = \frac{f_i(z_{i-1,k+1} + \mu_i u_{i,k+1}) - f_i(z_{i-1,k+1})}{\mu_i} u_{i,k+1}.$$

Step 2.2: Update

$$z_{i,k+1} = \Pi_X[z_{i-1,k+1} - \alpha_{k+1}\tilde{g}(z_{i-1,k+1}, u_{i,k+1})]. \tag{4}$$

End.

Step 3: k=k+1, go to Step 1.

Step 4: Until a predefined stopping criterion is met. Output x_k.

In the algorithm, the vector x_k is the estimate at the end of cycle k, $z_{i,k+1}$ is the intermediate estimate obtained after agent i updates in $(k+1)st$ cycle. In addition, $u_{i,k+1}$ is assumed as an i.i.d. random vector, Π_X denotes Euclidean projection onto the set X.

In terms of the updates (3) and (4), the algorithm CIGF generates a random sequence $\{x_k\}_{k\geq 1}$. Denote the σ-field $\mathcal{U}_{k+1}^i$ generated by the entire history of the random variables $u_{j,l}$ to iterations $(k+1)$, i.e., $\mathcal{U}_{k+1}^i = \sigma\{(u_{j,l}, j = 1, \ldots, m,\ l = 1, \ldots, k);\ (u_{j,k+1}, j = 1, \ldots, i)\},\ i = 1, \ldots, m$, and let $\mathcal{U}_k^m = \mathcal{U}_{k+1}^0$.

For any stepsize rules, we firstly establish a lemma to reveal a basic relation for the iterates generated by the algorithm CIGF, which plays a key role in our subsequent analysis. All the results and proofs in this paper parallel those of the incremental subgradient method, see e.g., [17, 4]. The essential difference is that we only use the evaluation of the agents' function values rather than the subgradients.

Lemma 3.1. *Let $\{x_k\}_{k\geq 1}$ be the sequence generated by the algorithm CIGF. For any non-increasing sequence $\{\alpha_k\}_{k\geq 1}$ of positive stepsizes, then, we have*

$$\begin{aligned} E[||x_{k+1} - y||^2 | \mathcal{U}_k^m] \leq\ & ||x_k - y||^2 - 2\alpha_{k+1}(F(x_k) - F(y)) + \alpha_{k+1}^2 \beta_1 m^2 L^2 \\ & + 2\alpha_{k+1}\mu\sqrt{d} m L, \end{aligned} \tag{5}$$

where $\beta_1 = d + (d+4)^2/m$ and $\mu = \max_{1\leq i\leq m}\{\mu_i\}$.

Proof. Using the iterate update (4) and the nonexpansion property of the projection Π_X, we have for any $y \in X$,

$$\begin{aligned} ||z_{i,k+1} - y||^2 &= ||\Pi_X[z_{i-1,k+1} - \alpha_{k+1}\tilde{g}_{i,k+1}] - y||^2 \\ &\leq ||(z_{i-1,k+1} - y) - \alpha_{k+1}\tilde{g}_{i,k+1}||^2 \\ &= ||z_{i-1,k+1} - y||^2 - 2\alpha_{k+1}\tilde{g}_{i,k+1}^T(z_{i-1,k+1} - y) + \alpha_{k+1}^2||\tilde{g}_{i,k+1}||^2. \end{aligned}$$

Taking conditional expectations with respect to the σ-field $\mathcal{U}_{k+1}^{i-1}$, we further obtain

$$\begin{aligned} E[||z_{i,k+1} - y||^2 | \mathcal{U}_{k+1}^{i-1}] \leq\ & ||z_{i-1,k+1} - y||^2 - 2\alpha_{k+1} E[\tilde{g}_{i,k+1} | \mathcal{U}_{k+1}^{i-1}]^T (z_{i-1,k+1} - y) \\ & + \alpha_{k+1}^2 E[||\tilde{g}_{i,k+1}||^2 | \mathcal{U}_{k+1}^{i-1}]. \end{aligned}$$

Noting that the fact $E[\tilde{g}(z_{i-1,k+1}, u_{i,k+1})|\mathcal{U}_{k+1}^{i-1}] = \nabla f_i^{\mu_i}$ obtained from Lemma 2.1 (iii) and $\mu_i \leq \mu$, and letting $f = f_i, x = z_{i-1,k+1}, \bar{x} = y$ in Lemma 2.2, the above relation yields

$$\begin{aligned} E[||z_{i,k+1} - y||^2|\mathcal{U}_{k+1}^{i-1}] &\leq ||z_{i-1,k+1} - y||^2 - 2\alpha_{k+1}(f_i(z_{i-1,k+1}) - f_i(y)) \\ &\quad +\alpha_{k+1}^2(d+4)^2L^2 + 2\alpha_{k+1}\mu\sqrt{d}L. \end{aligned}$$

Taking now the expectation conditional on $\mathcal{U}_k^m$, we get

$$\begin{aligned} E[||z_{i,k+1} - y||^2|\mathcal{U}_k^m] &\leq E[||z_{i-1,k+1} - y||^2|\mathcal{U}_k^m] - 2\alpha_{k+1}E[f_i(z_{i-1,k+1}) \\ &\quad -f_i(y)|\mathcal{U}_k^m] + \alpha_{k+1}^2(d+4)^2L^2 + 2\alpha_{k+1}\mu\sqrt{d}L. \end{aligned}$$

Summing over $i = 1, \ldots, m$, we have for any $y \in X$,

$$\begin{aligned} & E[||x_{k+1} - y||^2|\mathcal{U}_k^m] \\ \leq\ & ||x_k - y||^2 - 2\alpha_{k+1}\textstyle\sum_{i=1}^m E[f_i(z_{i-1,k+1}) - f_i(y)|\mathcal{U}_k^m] \\ & +\alpha_{k+1}^2(d+4)^2mL^2 + 2\alpha_{k+1}\mu\sqrt{d}mL \\ =\ & ||x_k - y||^2 - 2\alpha_{k+1}(F(x_k) - F(y)) - 2\alpha_{k+1}\textstyle\sum_{i=1}^m 2E[f_i(z_{i-1,k+1}) \\ & -f_i(x_k)|\mathcal{U}_k^m] + \alpha_{k+1}^2(d+4)^2mL^2 + 2\alpha_{k+1}\mu\sqrt{d}mL. \end{aligned}$$

We now estimate the term $\sum_{i=1}^m 2E[f_i(z_{i-1,k+1}) - f_i(x_k)|\mathcal{U}_k^m]$ in the preceding relation. By using Assumption 1, Lemma 2.1 and the iterate updates (3) and (4), we have

$$\begin{aligned} & \textstyle\sum_{i=1}^m E[f_i(z_{i-1,k+1}) - f_i(x_k)|\mathcal{U}_k^m] \leq \sum_{i=1}^m L\ E[||z_{i-1,k+1} - x_k||\ |\mathcal{U}_k^m] \\ & \leq L\textstyle\sum_{i=1}^m E[||\sum_{j=1}^{i-1}(z_{j,k+1} - z_{j-1,k+1})||\ |\mathcal{U}_k^m] \leq L\sum_{i=1}^m (i-1)\alpha_{k+1}dL \\ & = \alpha_{k+1}\tfrac{m(m-1)}{2}dL^2 \leq \alpha_{k+1}\tfrac{m^2}{2}dL^2. \end{aligned}$$

By combining the preceding relations and letting $\beta_1 = d + (d+4)^2/m$, we can obtain the desired result. □

We first consider the case with a constant stepsize rule.

Theorem 3.2. *Let $\{x_k\}_{k\geq 1}$ be the sequence generated by the algorithm CIGF, with a constant stepsize rule, i.e., $\alpha_k = \alpha > 0$ for all $k \geq 1$. Then, we have*

$$\liminf_{k\to\infty} E[F(x_k)] \leq F^* + \frac{\alpha\beta_1 m^2L^2}{2} + \mu\sqrt{d}mL. \tag{6}$$

Proof. By contradiction, suppose that the result of the theorem does not hold, there exists an $\epsilon > 0$ and an index $k_\epsilon > 0$ such that for all $k \geq k_\epsilon$,

$$E[F(x_k)] > F^* + \frac{\alpha\beta_1 m^2L^2}{2} + \mu\sqrt{d}mL + \epsilon.$$

Letting $y = x^*, x^* \in X^*, \alpha_{k+1} = \alpha$ in (5) and then taking all expectations, this implies

$$E[||x_{k+1} - x^*||^2] \leq E[||x_k - x^*||^2] - 2\alpha(E[F(x_k)] - F^*) + \alpha^2\beta_1 m^2L^2 + 2\alpha\mu\sqrt{d}mL.$$

By combining the above relations, we have

$$E[||x_{k+1} - x^*||^2] \leq E[||x_{k_\epsilon} - x^*||^2] - 2\epsilon(k - k_\epsilon),$$

which cannot hold for k sufficiently large. Hence, (6) must hold. □

Let K represent the number of cycles. The following theorem provides an estimate of K, required to reach a given level of optimality up to an error tolerance. Let the notation $\lceil a \rceil$ stand for the smallest integer greater than or equal to $a \in \mathbb{R}$.

Theorem 3.3. *Let $\{x_k\}_{k\geq 1}$ be the sequence generated by the algorithm CIGF, with a constant stepsize rule, i.e., $\alpha_k = \alpha > 0$ for all $k \geq 1$. Then, for any $\epsilon > 0$ and $x^* \in X^*$, we have*

$$\min_{1\leq k\leq K} E[F(x_k)] \leq F^* + \frac{\alpha\beta_1 m^2 L^2}{2} + \mu\sqrt{d}mL + \frac{\epsilon}{3}, \tag{7}$$

where $K = \lceil 3||x_0 - x^*||^2/(2\alpha\epsilon)\rceil$.

Proof. Suppose that (7) dose not hold, then for all k with $1 \leq k \leq K$, we have

$$E[F(x_k)] > F^* + \frac{\alpha\beta_1 m^2 L^2}{2} + \mu\sqrt{d}mL + \frac{\epsilon}{3}.$$

Letting $y = x^*, \alpha_{k+1} = \alpha$ in (5) and taking all expectations, give rise to

$$E[||x_{k+1} - x^*||^2] \leq E[||x_k - x^*||^2] - \frac{2}{3}\alpha\epsilon,$$

Summation of the above inequalities over k for $k = 0, \ldots, K$, gives

$$E[||x_{K+1} - x^*||^2] \leq ||x_0 - x^*||^2 - \frac{2(K+1)}{3}\alpha\epsilon,$$

which contradicts to the definition of K. □

Remark 1. According to the iterates (3) and (4), every cycle requires m sub-iterations, so the total number N of component functions that must be evaluated in order for satisfying (7) is given by $N = mK = m\lceil 3||x_0 - x^*||^2/(2\alpha\epsilon)\rceil$. In Theorem 3.3, for any given $\epsilon > 0$, if we choose the smoothing parameter μ and the constant stepsize α satisfied:

$$\mu \leq \epsilon/(3\sqrt{d}mL),\ \alpha \leq 2\epsilon/(3\beta_1 m^2 L^2), \tag{8}$$

we can achieve $\min_{1\leq k\leq K} E[F(x_k)] - F^* \leq \epsilon$. This implies that the total number of necessary iterations is at most

$$N(\epsilon, d, m) = \mathcal{O}(d^2 m^3 L^2/\epsilon^2). \tag{9}$$

Note that the iteration complexity bound (9) of the proposed algorithm CIGF is in $\mathcal{O}(d^2)$ times worse than that of the cyclic incremental subgradient method proposed in [16, 17, 4]. This can be explained by the upper bound $\mathbb{E}[||\tilde{g}_i||] \leq dL$ provided by Lemma 2.1 (iv), which is different from the subgradient upper bound $||g_i|| \leq L, \forall g_i \in \partial f_i$ provided in [16, 17, 4]. However, our method only requires the evaluation of the function values rather than subgradients. When $m = 1$, the method reduces to the case considered in [19].

We now consider a convergence result for a diminishing stepsize case.

Theorem 3.4. *Let $\{x_k\}_{k\geq 1}$ be the sequence generated by the algorithm CIGF, with a diminishing stepsize rule satisfied $\alpha_k > 0$, $\lim_{k\to\infty}\alpha_k = 0$ and $\sum_{k=1}^{\infty}\alpha_k = \infty$. Then,*

$$\liminf_{k\to\infty} E[F(x_k)] = F^* + \mu\sqrt{d}mL. \tag{10}$$

Proof. By contradiction, suppose that (10) does not hold, then there exists an $\epsilon > 0$ and an index $k_\epsilon > 0$ such that for all $k \geq k_\epsilon$,

$$E[F(x_k)] - F^* - \mu\sqrt{d}mL > \epsilon.$$

Letting $y = x^*, x^* \in X^*$ in (5) and taking all expectations, we obtain

$$E[||x_{k+1} - x^*||^2] \le E[||x_k - x^*||^2] - 2\alpha_{k+1}(E[F(x_k)] - F^*) + \alpha_{k+1}^2\beta_1 m^2 L^2 + 2\alpha_{k+1}\mu\sqrt{d}mL.$$

Combining the preceding relations leads to

$$E[||x_{k+1} - x^*||^2] \le E[||x_k - x^*||^2] - 2\alpha_{k+1}\epsilon + \alpha_{k+1}^2\beta_1 m^2 L^2.$$

Since $\lim_{k\to\infty} \alpha_k = 0$, without loss of generality, we may assume that k_ϵ is large enough such that

$$\alpha_{k+1} \le \epsilon/(\beta_1 m^2 L^2), \ \forall k \ge k_\epsilon.$$

Thus, for all $k \ge k_\epsilon$, we have

$$E[||x_{k+1} - x^*||^2] \le E[||x_k - x^*||^2] - \alpha_{k+1}\epsilon \le \cdots \le E[||x_{k_\epsilon} - x^*||^2] - \epsilon \sum_{l=k_\epsilon}^{k} \alpha_{l+1},$$

which cannot hold for k sufficiently large due to the condition $\sum_{k=1}^{\infty} \alpha_k = \infty$. Hence, (10) holds. □

4. **Incremental gradient-free method with randomized order.** In this section, based on the randomized incremental subgradient method developed in [17, 4], we first propose a randomized incremental gradient-free algorithm, then give the convergence analysis of the algorithm under different stepsize rules. By comparison with the previous algorithm CIGF, the algorithm developed in this section is applicable to a broader class of networks.

Now we assume that the network of agents is fully connected [22], and develop an incremental algorithm where the agent (only known its component function value) that updates is selected randomly at each iteration over the network. Formally, the randomized incremental gradient-free (RIGF, for short) algorithm is given as follows:

Algorithm RIGF

Initialization: Given an initial point $x_0 \in X$, stepsizes $\{\alpha_{k+1}\}_{k\ge0}$, smoothing parameters $\mu_{k+1} > 0$. Set $k = 0$.

Step 1: Generate uniformly ω_{k+1} from the set of agents $\{1, \ldots, m\}$ and choose the corresponding component function $f_{\omega_{k+1}}$.

Step 2: Generate u_{k+1} by Gaussian random vector generator and call the gradient-free oracle of the component function $f_{\omega_{k+1}}$ for computing $\tilde{g}(x_k, \omega_{k+1}, u_{k+1})$ given by

$$\tilde{g}(x_k, \omega_{k+1}, u_{k+1}) = \frac{f_{\omega_{k+1}}(x_k + \mu_{k+1}u_{k+1}) - f_{\omega_{k+1}}(x_k)}{\mu_{k+1}} u_{k+1}.$$

Step 3: Update

$$x_{k+1} = \Pi_X[x_k - \alpha_{k+1}\tilde{g}(x_k, \omega_{k+1}, u_{k+1})]. \tag{11}$$

Step 4: k=k+1, go to Step 1.

Step 5: Until a predefined stopping criterion is met. Output x_k.

We assume that: 1) $\{\omega_{k+1}\}$ is a sequence of independent random variables, which is independent of the sequence $\{x_{k+1}\}$ [17]; 2) $\{u_{k+1}\}$ is a sequence of i.i.d. random vectors; 3) the sequences $\{\omega_{k+1}\}$ and $\{u_{k+1}\}$ are independent. Denote the σ-field $\mathcal{F}_k = \sigma\{(\omega_j, u_j)|j = 1, \ldots, k\}$ generated by the entire history of the random

variables ω_j and u_j up to iterations k. For the simplified notation, we also assume that $\mu_{k+1} \leq \mu$ for all k with $\mu > 0$.

We first deal with the case of a constant stepsize.

Theorem 4.1. *Let $\{x_k\}_{k\geq 1}$ be the sequence generated by the algorithm RIGF, with a constant stepsize rule, i.e., $\alpha_k = \alpha > 0$ for all $k \geq 1$. Then, with probability 1, we have*

$$\inf_{k\geq 1} F(x_k) - F^* \leq \frac{\alpha\beta_2 mL^2}{2} + \mu\sqrt{d}mL. \tag{12}$$

where $\beta_2 = (d+4)^2$.

Proof. Using the update (11) and the nonexpansion property of the projection Π_X, we have for any $y \in X$,

$$\begin{aligned} ||x_{k+1} - y||^2 &\leq ||x_k - \alpha_{k+1}\tilde{g}(x_k, \omega_{k+1}, u_{k+1}) - y||^2 \\ &= ||x_k - y||^2 - 2\alpha_{k+1}\tilde{g}(x_k, \omega_{k+1}, u_{k+1})^T(x_k - y) \\ &\quad +\alpha_{k+1}^2||\tilde{g}(x_k, \omega_{k+1}, u_{k+1})||^2. \end{aligned}$$

Taking the expectations with respect to $\mathcal{F}_k$ in the above inequality, we can obtain

$$\begin{aligned} &E\left[||x_{k+1} - y||^2|\mathcal{F}_k\right] \\ &\leq ||x_k - y||^2 - 2\alpha_{k+1}E[\tilde{g}(x_k, \omega_{k+1}, u_{k+1})|\mathcal{F}_k]^T(x_k - y) \\ &\quad +\alpha_{k+1}^2 E[||\tilde{g}(x_k, \omega_{k+1}, u_{k+1})||^2|\mathcal{F}_k]. \end{aligned}$$

By Lemma 2.1 and the assumption that both ω_k and u_k are independent, we have

$$\begin{aligned} E[||\tilde{g}(x_k, \omega_{k+1}, u_{k+1})||^2|\mathcal{F}_k] &= E_{\omega_k}[E_{u_k}[||\tilde{g}(x_k, \omega_{k+1}, u_{k+1})||^2]] \\ &\leq E_{\omega_k}[(d+4)^2L^2] = (d+4)^2L^2. \end{aligned}$$

In addition, letting $f = f_{\omega_{k+1}}, x = x_k, \bar{x} = y, \nu = \mu_{k+1}$ in Lemma 2.2, then using the relation $E[\tilde{g}(x_k, \omega_{k+1}, u_{k+1})|\mathcal{F}_k] = \nabla f_{\omega_{k+1}}^{\mu_{k+1}}$ obtained from Lemma 2.1 and $\mu_{k+1} \leq \mu$, we have

$$E[f_{\omega_{k+1}}(x_k) - f_{\omega_{k+1}}(y)|\mathcal{F}_k] \leq E[\tilde{g}(x_k, \omega_{k+1}, u_{k+1})|\mathcal{F}_k](x_k - y) + \mu\sqrt{d}L.$$

Combining the preceding relations gives rise to

$$\begin{aligned} &E[||x_{k+1} - y||^2|\mathcal{F}_k] \\ \leq\ &||x_k - y||^2 - 2\alpha_{k+1}E[f_{\omega_{k+1}}(x_k) - f_{\omega_{k+1}}(y)|\mathcal{F}_k] + \alpha_{k+1}^2(d+4)^2L^2 \\ &+2\alpha_{k+1}\mu\sqrt{d}L \\ =\ &||x_k - y||^2 - 2\alpha_{k+1}\sum_{i=1}^m \frac{1}{m}(f_i(x_k) - f_i(y)) + \alpha_{k+1}^2(d+4)^2L^2 + 2\alpha_{k+1}\mu\sqrt{d}L \\ =\ &||x_k - y||^2 - \frac{2\alpha_{k+1}}{m}(F(x_k) - F(y)) + \alpha_{k+1}^2\beta_2 L^2 + 2\alpha_{k+1}\mu\sqrt{d}L, \end{aligned} \tag{13}$$

where in the first equality we use the fact that ω_{k+1} takes the values $1, \ldots, m$ with equal probability $1/m$, in the second equality we let $\beta_2 = (d+4)^2$. Similar to the proof of Proposition 3.1 in [17], for a fixed positive scalar γ, we construct the following level set:

$$L_\gamma = \{x \in X \mid F(x) < F^* + \frac{2}{\gamma} + \frac{\alpha\beta_2 mL^2}{2} + \mu\sqrt{d}mL\},$$

and let $y_\gamma \in X$ be such that $F(y_\gamma) = F^* + \frac{1}{\gamma}$. Obliviously, $y_\gamma \in L_\gamma$ by construction. Define a sequence $\{\hat{x}_k\}$ as follows:

$$\hat{x}_{k+1} = \begin{cases} x_{k+1}, & \text{if } \hat{x}_k \notin L_\gamma, \\ y_\gamma, & \text{otherwise.} \end{cases}$$

Thus, the process $\{\hat{x}_k\}$ is identical to the process $\{x_k\}$, except that once x_k enters the level set L_γ, the process terminates with $\hat{x}_k = y_\gamma$. Letting $y = y_\gamma$ and $\alpha_{k+1} = \alpha$ in (13), we have

$$\begin{aligned} E[||\hat{x}_{k+1} - y_\gamma||^2 | \mathcal{F}_k] &\leq ||\hat{x}_k - y_\gamma||^2 - \frac{2\alpha}{m}(F(\hat{x}_k) - F(y_\gamma)) + \alpha^2\beta_2 L^2 + 2\alpha\mu\sqrt{d}L \\ &\leq ||\hat{x}_k - y_\gamma||^2 - \xi_{k+1}, \end{aligned} \tag{14}$$

where

$$\xi_{k+1} = \begin{cases} \frac{2\alpha}{m}(F(\hat{x}_k) - F(y_\gamma)) - \alpha^2\beta_2 L^2 - 2\alpha\mu\sqrt{d}L, & \text{if } \hat{x}_k \notin L_\gamma, \\ 0, & \text{if } \hat{x}_k = y_\gamma. \end{cases}$$

Now we show that $\xi_{k+1} \geq 0$. If $\hat{x}_k \notin L_\gamma$, by the definition of ξ_{k+1}, we have

$$\xi_{k+1} \geq \frac{2\alpha}{m}(F^* + \frac{2}{\gamma} + \frac{\alpha\beta_2 m L^2}{2} + \mu\sqrt{d}mL - F^* - \frac{1}{\gamma}) - \alpha^2\beta_2 L^2 - 2\alpha\mu\sqrt{d}L = \frac{2\alpha}{m\gamma}.$$

Hence, $\xi_{k+1} \geq 0$ for all k. By (14) and the Supermartingale Convergence Theorem (see, Proposition 2 in [5]), with probability 1, we obtain $\sum_{k=0}^{\infty} \xi_{k+1} \leq \infty$, which implies that $\hat{x}_k \in L_\gamma$ for sufficiently large k. By letting $\gamma \to \infty$, we can obtain (12). □

Next we obtain an estimate on the expected number of iterations for algorithm RIGF, which parallels Theorem 3.3 for the non-randomized case.

Theorem 4.2. *Let $\{x_k\}_{k\geq 1}$ be the sequence generated by the algorithm RIGF, with a constant stepsize rule, i.e., $\alpha_k = \alpha > 0$ for all $k \geq 1$. Then, for any $\epsilon > 0$ and $x^* \in X^*$, with probability 1, we have,*

$$\min_{1\leq k\leq N} F(x_k) - F^* \leq \frac{\alpha\beta_2 m L^2}{2} + \mu\sqrt{d}mL + \frac{\epsilon}{3}, \tag{15}$$

where N is a random variable with $E[N] \leq 3m||x_0 - x^*||^2/(2\alpha\epsilon)$.

Proof. Let $\hat{y} \in X^*$ be some fixed vector. Define a new process $\{\hat{x}_k\}$ which is identical to $\{x_k\}$, except that once x_k enters the level set

$$L = \{x \in X \mid F(x) < F^* + \frac{\alpha\beta_2 m L^2}{2} + \mu\sqrt{d}mL + \frac{\epsilon}{3}\},$$

the process $\{\hat{x}_k\}$ terminates at $\hat{y}$. Similar to the proof of Theorem 4.1 (cf. Eq. (13)), for the process $\{\hat{x}_k\}$, we obtain for all k

$$\begin{aligned} E[||\hat{x}_{k+1} - x^*||^2 | \mathcal{F}_k] &\leq ||\hat{x}_k - x^*||^2 - \frac{2\alpha}{m}(F(\hat{x}_k) - F^*) + \alpha^2\beta_2 L^2 + 2\alpha\mu\sqrt{d}L \\ &= ||\hat{x}_k - x^*||^2 - \eta_{k+1}, \end{aligned} \tag{16}$$

where

$$\eta_{k+1} = \begin{cases} \frac{2\alpha}{m}(F(\hat{x}_k) - F^*) - \alpha^2\beta_2 L^2 - 2\alpha\mu\sqrt{d}L, & \text{if } \hat{x}_k \notin L, \\ 0, & \text{otherwise.} \end{cases}$$

In the case where $\hat{x}_k \notin L$, using the definitions of L and η_{k+1}, we have

$$\eta_{k+1} \geq \frac{2\alpha}{m}(F^* + \frac{\alpha\beta_2 mL^2}{2} + \mu\sqrt{d}mL + \frac{\epsilon}{3} - F^*) - \alpha^2\beta_2 L^2 - 2\alpha\mu\sqrt{d}L = \frac{2\alpha\epsilon}{3m}. \quad (17)$$

By the Supermartingale Convergence Theorem (see, Proposition 2 in [5]), from (16) we obtain $\sum_{k=0}^{\infty}\xi_{k+1} < \infty$ with probability 1, so that $\eta_{k+1} = 0$ for all $k \geq N$, where N is a random variable. Hence $\hat{x}_N \in L$ with probability 1, implying that in the original process we have

$$\min_{1\leq k\leq N} F(x_k) \leq F^* + \frac{\alpha\beta_2 mL^2}{2} + \mu\sqrt{d}mL + \frac{\epsilon}{3},$$

with probability 1. Furthermore, by taking the total expectation in (16), we obtain for all k,

$$E[||\hat{x}_{k+1} - x^*||^2] \leq E[||\hat{x}_k - x^*||^2] - E[\eta_{k+1}] \leq ||x_0 - x^*||^2 - E[\sum_{j=0}^{k}\eta_{j+1}],$$

where in the last inequality we use the facts $\hat{x}_0 = x_0$ and $E[||\hat{x}_0 - x^*||^2] = ||\hat{x}_0 - x^*||^2$. Therefore, letting $k \to \infty$, and using the definition of η_{k+1} and the relation (17), we have

$$||x_0 - x^*||^2 \geq E[\sum_{j=0}^{\infty}\eta_{j+1}] = E[\sum_{j=0}^{N-1}\eta_{j+1}] \geq E[\frac{2N\alpha\epsilon}{3m}] = \frac{2\alpha\epsilon}{3m}E[N].$$

□

Remark 2. For any given $\epsilon > 0$, if we choose the smoothing parameter μ and the constant stepsize α in Theorem 4.2 such that:

$$\mu \leq \epsilon/(3\sqrt{d}mL),\ \alpha \leq 2\epsilon/(3\beta_2 mL^2),$$

we can achieve $\min_{1\leq k\leq N} F(x_k) - F^* \leq \epsilon$, which implies that the expected number of iterations is at most

$$E[N(\epsilon, d, m)] = \mathcal{O}(d^2m^2L^2/\epsilon^2). \quad (18)$$

By compared the bounds (18) with (9), the algorithm RIGF is much faster than the algorithm CIGF (a factor of m) in the sense of expectation. In addition, due to the replacement of subgradient by using only the evaluation of the objective function values, the iteration complexity of our algorithm RIGF is in $\mathcal{O}(d^2)$ times worse than that of the randomized incremental subgradient method proposed in [16, 17, 4].

In parallel with the result of Theorem 3.4, we give the following convergence result for a diminishing stepsize case.

Theorem 4.3. *Let $\{x_k\}_{k\geq 1}$ be the sequence generated by the algorithm RIGF, with a diminishing stepsize rule satisfied $\alpha_k > 0$, $\lim_{k\to\infty}\alpha_k = 0$ and $\sum_{k=1}^{\infty}\alpha_k = \infty$. Then, we have*

$$\liminf_{k\to\infty} E[F(x_k)] = F^* + \mu\sqrt{d}mL. \quad (19)$$

Proof. To arrive at a contradiction, assume that there exists an $\epsilon > 0$ and an integer $k_1 > 0$ such that

$$E[F(x_k)] - F^* - \mu\sqrt{d}mL > \epsilon, \forall k \geq k_1.$$

Letting $y = x^*, x^* \in X^*$ in (13) and taking all expectations, we have

$$\begin{aligned} E[||x_{k+1} - x^*||^2] &\leq E[||x_k - x^*||^2] - 2\alpha_{k+1}(E[F(x_k)] - F^*) \\ &\quad + \alpha_{k+1}^2 \beta_2 m^2 L^2 + 2\alpha_{k+1}\mu\sqrt{d}mL. \end{aligned}$$

Since $\lim_{k\to\infty} \alpha_k = 0$, for the same ϵ as above, there exists an integer $k_2 > 0$ such that

$$\alpha_{k+1} \leq \epsilon/(\beta_2 m^2 L^2), \ \forall k \geq k_2.$$

By taking $k_0 = \max\{k_1, k_2\}$, thus, for all $k \geq k_0$, we obtain

$$E[||x_{k+1} - x^*||^2] \leq E[||x_k - x^*||^2] - \epsilon\alpha_{k+1} \leq \cdots \leq E[||x_{k_0} - x^*||^2] - \epsilon \sum_{l=k_0}^{k} \alpha_{l+1},$$

which cannot hold for k sufficiently large due to the condition $\sum_{k=1}^{\infty} \alpha_k = \infty$. Hence, (19) holds. □

5. **Numerical simulation.** In this section, we illustrate some experimental results on the convergence behaviors of the proposed incremental gradient-free algorithms as a function of number of agents m as well as the dimension of the agent d. The comparisons among our algorithms, the cyclic incremental subgradient (CISG, for short) algorithm and the randomized incremental subgradient (RISG, for short) algorithm proposed in [17] under the same stepsize updating rules are also presented.

Consider a robust linear l_1-regression problem commonly studied in system identification [20]. Specifically, given m pairs of the form $(a_i, b_i) \in \mathbb{R}^d \times \mathbb{R}$, we want to estimate a vector $x \in \mathbb{R}^d$ such that $a_i^T x \approx b_i$. The linear l_1-regression problem can be formulated as follows:

$$\min_{x\in\mathbb{R}^d} F(x) := \sum_{i=1}^{m} |a_i^T x - b_i|, \ \text{s.t.} \ ||x|| \leq R, \tag{20}$$

where $||x|| \leq R$ is the l_2 norm constraint. Clearly, $f_i(x) = |a_i^T x - b_i|$ is non-differentiable at any point with $a_i^T x = b_i$. However, $f_i(x)$ is convex and L-Lipschitz by setting $L = \max_i ||a_i||$.

For a given network size m, we generate a random instance of a regression problem with m data points. In all tests, we set $R = 10$ and choose the parameter μ that satisfies (8).

We first consider the constant stepsize case by setting $\alpha = 0.001$, dimensions of the agent $d = 1$ or 4, and number of agents $m = 100$ or 500. Fig. 1 depicts the value of $F(x_k) - F^*$ versus the number of cycles K by using algorithms CIGF and CISG, Fig. 2 plots the value of $F(x_k) - F^*$ versus the number of iterations N by using algorithms RIGF and RISG. From both figures, we can clearly see that all algorithms can achieve good convergence results. By comparison, we find that our algorithms CIGF and RIGF can converge to a better suboptimal value than that of algorithms CISG and RISG, although algorithms CIGF and RIGF require more iterations than algorithms CISG and RISG. More importantly, our algorithms illustrate much smaller oscillation than algorithms CISG and RISG under the setting of the constant stepsize. This can be explained that our algorithms CIGF and RIGF are based on a smooth approximation of original objective functions, but subgradient-based algorithms CISG and RISG may frequently suffer from the case that $F(x_{k+1}) \not< F(x_k)$. Finally, in contrast to subgradient-based algorithms, our algorithms can carry out without the calculation of subgradient. They depend only

on the evaluations of function values, which are preferred by practical engineers. This is because for nonsmooth optimization problems, some substantial efforts of the computation of the subgradient require a certain knowledge of convex analysis.

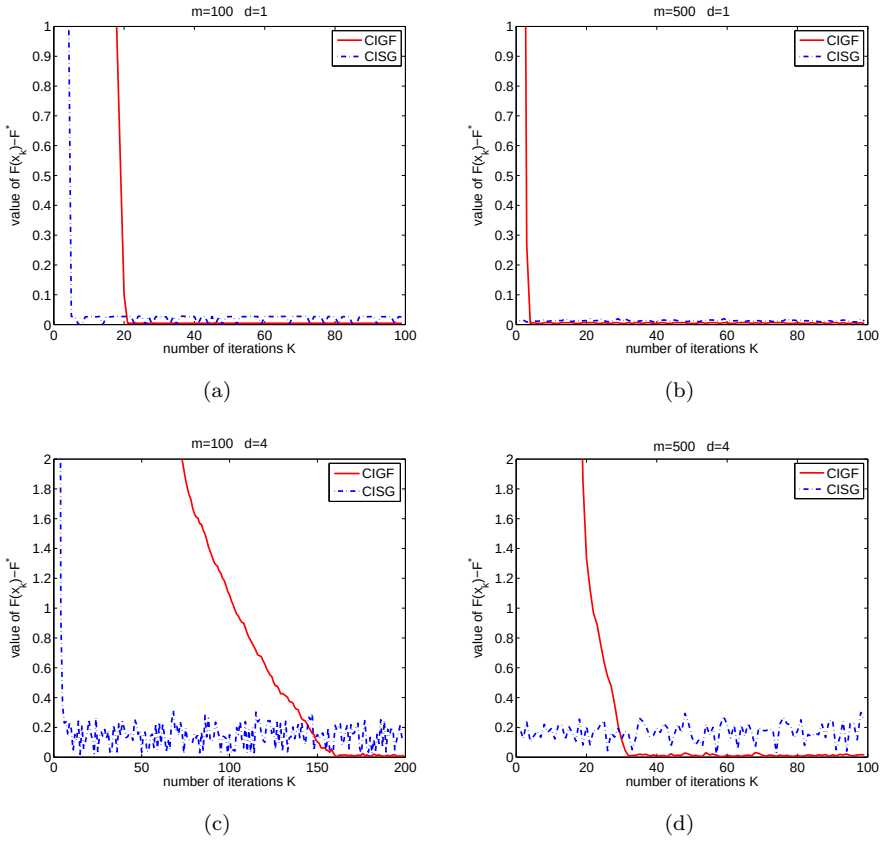


FIGURE 1. Function value error versus number of cycles K with a constant stepsize $\alpha = 0.001$ for algorithms CIGF and CISG

To illustrate the incremental gradient-free method with the diminishing stepsize rule, we fix $m = 100$ and $d = 2$, and consider two diminishing stepsize cases as $1/k$ and $1/k^{2/3}$. Fig. 3 plots the value of $F(x_k) - F^*$ versus number of iterations under these diminishing stepsize choices. It can been observed from Fig. 3 that all algorithms can achieve the good convergence, but all of them are sensitive to the choice of diminishing stepsizes. The performance of achieving the optimal value for algorithms chosen the stepsize $1/k$ is much better that of algorithms chosen the stepsize $1/k^{2/3}$. Under the choice of the stepsize $1/k$, the convergence performance of our algorithms CIGF and RIGF is slightly better than that of algorithms CISG and RISG.

In Fig. 4, we present the actual behaviors of algorithm CIGF related to the dimension of agents d and number of agents m with a pre-fixed target accuracy $\epsilon = 0.01$ and a constant stepsize $\alpha = 0.001$. In each panel, each point on the heavy red curve is the average of 20 trials for algorithm CIGF, on the dotted blue curve is the average of 20 trials for algorithm CISG, and the vertical bars are the corresponding

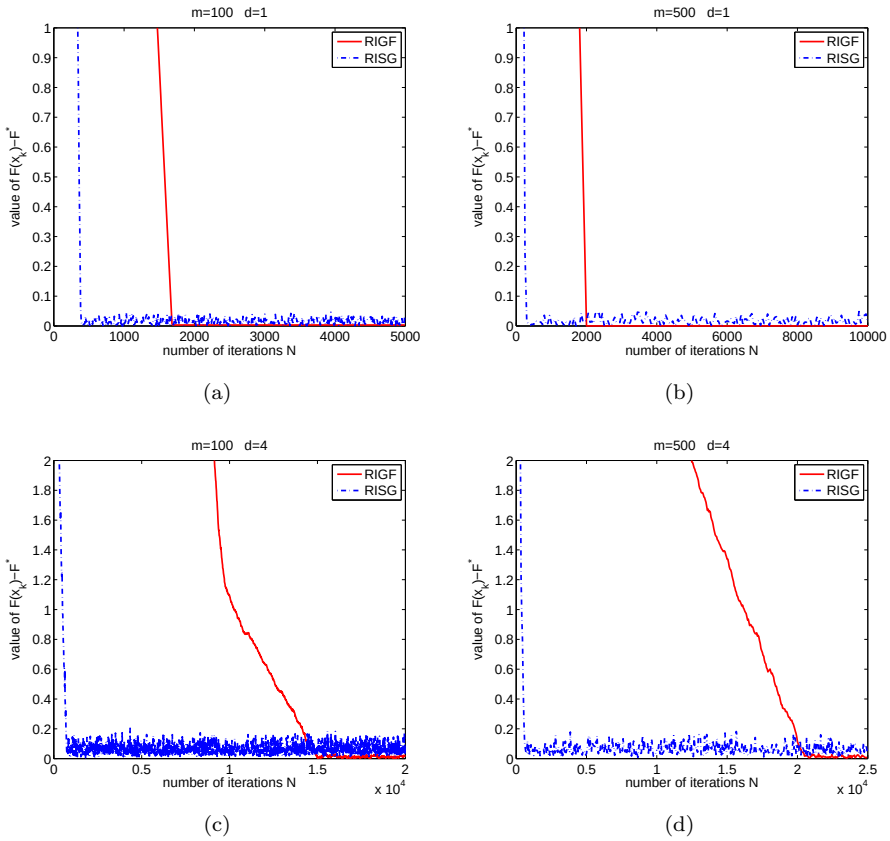
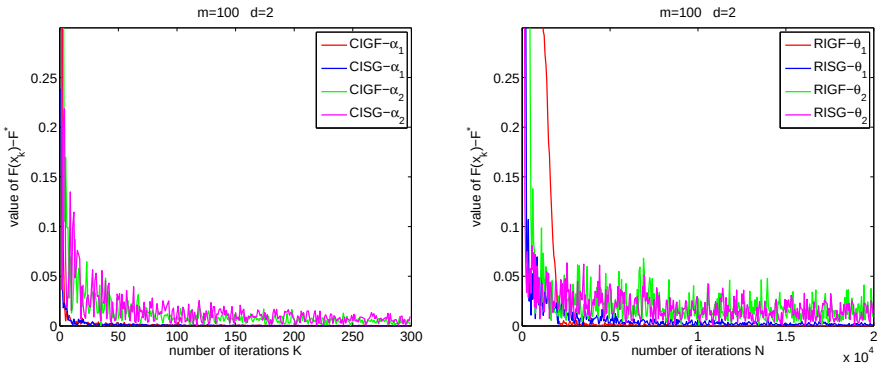


FIGURE 2. Function value error versus number of iterations N with a constant stepsize $\alpha = 0.001$ for algorithms RIGF and RISG

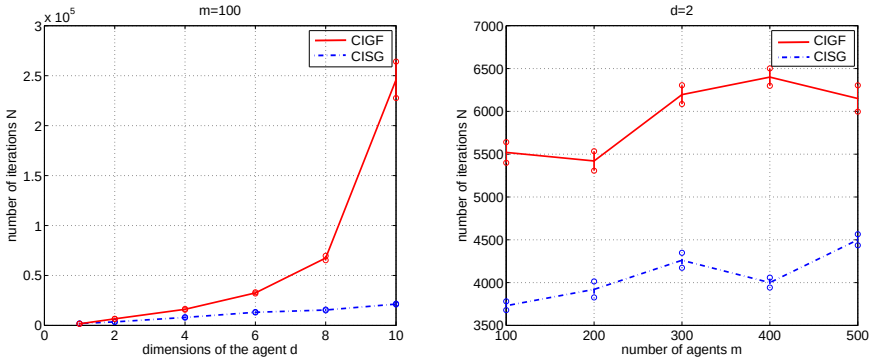
standard errors. Fig. 4(a) shows the value of $N(\epsilon, d, m)$ for algorithm CIGF versus dimensions of the agent for a fixed number of agents $m = 100$. The value of $N(\epsilon, d, m)$ increases dramatically as d increases when compared with the number of iterations required by algorithm CISG. This is because $N(\epsilon, d, m)$ obtained in (9) for the fixed ϵ and m, which is at most $\mathcal{O}(d^2)$ times worse than that of algorithm CISG. Fig. 4(b) depicts the value of $N(\epsilon, d, m)$ versus the number of agents m for a fixed dimension of the agent $d = 2$. In Fig. 4(b), some small fluctuations on the number of iterations for both algorithms arise as m increases. The number of iterations $N(\epsilon, d, m)$ of algorithm CIGF is slightly greater than that of algorithm CISG. This is because we fix the dimension of the agent $d = 2$, thus, the number of iterations for algorithm CIGF is at most $d^2 = 4$ times worse than that of algorithm CISG according to the given estimate given (9) in theory. These results show the excellent agreement of the empirical behavior with our theoretical predictions.

6. **Conclusions.** In this paper, based on the framework of the incremental approach, we have proposed incremental gradient-free methods for minimizing the sum of many-terms local convex functions embedded in multi-agent networks. We have proved the convergence of the proposed algorithms and provided the iteration



(a) (b)

FIGURE 3. (a) Function value error versus number of cycles K with diminishing stepsize choices: $\alpha_1(k) = 1/(m(k-1)+i)$, $\alpha_2(k) = 1/(m(k-1)+i)^{\frac{2}{3}}, k = 0, 1, \ldots, i = 1, \ldots, m$; (b) Function value error versus number of iterations N with diminishing stepsize choices: $\theta_1(k) = 1/k$, $\theta_2(k) = 0.1/k^{\frac{2}{3}}, k = 1, 2, \ldots$.



(a) (b)

FIGURE 4. For a fixed target accuracy $\epsilon = 0.01$ and a constant stepsize $\alpha = 0.001$, comparisons between algorithms CIGF and CISG: (a) number of iterations N versus dimensions of the agent d for fixed $m = 100$; (b) number of iterations N versus number of agents m for fixed $d = 2$.

complexity bounds as a function of the number of the agents m as well as the dimension of the problem d. A linear l_1-regression numerical example was used to show the excellent agreement of the empirical behavior with our theoretical predictions. In particular, our proposed methods depend only on the evaluations of the function value rather than subgradients, which may be preferred by practical engineers. Furthermore, for some nonsmooth problems, our methods can even achieve better numerical performance than that of subgradient-based methods.

Acknowledgements. We acknowledge the anonymous reviewers and the editor for their helpful feedback and thoughtful comments.

REFERENCES

[1] A. M. Bagirov, M. Ghosh and D. Webb, A derivative-free method for linearly constrained nonsmooth optimization, *J. Ind. Manag. Optim.*, **2** (2006), 319–338.
[2] D. P. Bertsekas, Stochastic optimization problems with nondifferentiable cost functionals, *J. Optim. Theory Appl.*, **12**(1973), 218–231.
[3] D. P. Bertsekas and J. N. Tsitsiklis, *Parallel and Distributed Computation: Numerical Methods*, Athena Scientific, Belmont, MA, 1989.
[4] D. P. Bertsekas, A. Nedić and E. Ozdaglar, *Convex Analysis and Optimization*, Athena Scientific, Belmont, MA, 2003.
[5] D. P. Bertsekas, Incremental proximal methods for large scale convex optimization, *Math. Program. B.*, **129** (2011), 163–195.
[6] A. R. Conn, K. Scheinberg and L. N. Vicente, *Introduction to Derivative-Free Optimization, MPS-SIAM Series on Optimization*, SIAM, Philadelphia, 2009.
[7] J. C. Duchi, A. Agarwal and M. J. Wainwright, Dual averaging for distributed optimization: Convergence analysis and network scaling, *IEEE Trans. Autom. Control.*, **57** (2012), 592–606.
[8] J. C. Duchi, P. L. Bartlet and M. J. Wainwrighr, Randomized smoothing for stochastic optimization, *SIAM J. Optim.*, **22** (2012), 674–701.
[9] X. X. Huang, X. Q. Yang and K. L. Teo, A smoothing scheme for optimization problems with Max-Min constraints, *J. Ind. Manag. Optim.*, **3** (2007), 209–222.
[10] J. Hiriart-Urruty and C. Lemarechal, *Convex Analysis and Minimization Algorithms I*, Springer, Berlin, 1996.
[11] X. Zhang, C. Wu, J. Li, X. Wang, Z. Yang, J. M. Lee and K. H. Jung, Binary artificial algae algorithm for multidimensional knapsack problems, *Applied Soft Computing*, **43** (2016), 583–595.
[12] B. Johansson, M. Rabi and M. Johansson, A randomized incremental subgradient method for distributed optimization in networked systems, *SIAM J. Optim.*, **20** (2009), 1157–1170.
[13] K. C. Kiwiel, Convergence of approximate and incremental subgradient methods for convex optimization, *SIAM J. Optim.*, **14** (2004), 807–840.
[14] J. Y. Li, C. Z. Wu, Z. Y. Wu and Q. Long, Gradient-free method for nonsmooth distributed optimization, *J. Glob. Optim.*, **61** (2015), 325–340.
[15] J. Y. Li, C. Z. Wu, Z. Y. Wu, Q. Long and X. Y. Wang, Distributed proximal-gradient method for convex optimization with inequality constraints, *ANZIAM J.*, **56** (2014), 160–178.
[16] A. Nedić and D. P. Bertsekas, Convergence rate of incremental subgradient algorithm, in *Stochastic Optimization: Algorithms and Applications* (eds. S. Uryasev and P. M. Pardalos), Applied Optimization, 54, Springer, 2001, 223–264.
[17] A. Nedić and D. P. Bertsekas, Incremental subgradient methods for nondifferentiable optimization, *SIAM J. Optim.*, **12** (2001), 109–138.
[18] A. Nedić and A. Ozdaglar, Distributed subgradient methods for multi-agent optimization, *IEEE Trans. Autom. Control.*, **54** (2009), 48–61.
[19] Y. Nesterov, *Random Gradient-Free Minimization of Convex Functions*, Technical report, Center for Operations Research and Econometrics (CORE), Catholic University of Louvain, January 2011. Available from: `http://www.ecore.be/DPs/dp_1297333890.pdf`.
[20] B. T. Polyak and J. Tsypkin, Robust identification, *Automatica*, **16** (1980), 53–63.
[21] M. G. Rabbat and R. D. Nowak, Quantized incremental algorithms for distributed optimization, *IEEE J. Sel. Areas Commun.*, **23** (2005), 798–808.
[22] S. S. Ram, A. Nedić and V. V. Veeravalli, Incremental stochastic subgradient algorithms for convex optimization, *SIAM J. Optim.*, **20** (2009), 691–717.
[23] Q. J. Shi, C. He and L. G. Jiang, Normalized incremental subgradient algorithm and its application, *IEEE Signal Processing*, **57** (2009), 3759–3774.
[24] R. L. Sheu, M. J. Ting and I. L. Wang, Maximum folw problem in the distribution network, *J. Ind. Manag. Optim.*, **2** (2006), 237–254.
[25] M. V. Solodov, Incremental gradient algorithms with stepsizes bounded away from zero, *Comput. Optim. Appl.*, **11** (1998), 28–35.
[26] D. M. Yuan, S. Y. Xu and J. W. Lu, Gradient-free method for distributed multi-agent optimization via push-sum algorithms, *Int. J. Robust Nonlinear Control*, **25** (2015), 1569–1580.

[27] Q. Long and C. Wu, A hybrid method combining genetic algorithm and Hooke-Jeeves method for constrained global optimization, *J. Ind. Manag. Optim.*, **10** (2014), 1279–1296.
[28] G. H. Yu, A derivative-free method for solving large-scale nonlinear systems of equations, *J. Ind. Manag. Optim.*, **6** (2010), 149–160.
[29] C. J. Yu, K. L. Teo, L. S. Zhang and Y. Q. Bai, A new exact penalty function method for continuous inequality constrained optimization problems, *J. Ind. Manag. Optim.*, **6** (2010), 895–910.
[30] F. Yousefian, A. Nedić and U. V. Shanbhag, On stochastic gradient and subgradient methods with adaptive steplength sequences, *Automatica*, **48** (2012), 56–67.
[31] J. Li, G. Chen, Z. Dong and Z. Wu, A fast dual proximal-gradient method for separable convex optimization with linear coupled constraints, *Comp. Opt. Appl.*, **64** (2016), 671–697.

Received March 2015; 1st revision January 2016; 2nd revision August 2016.

E-mail address: lijueyou@163.com
E-mail address: gqli2@163.com
E-mail address: zywu@cqnu.edu.cn
E-mail address: c.wu@curtin.edu.au
E-mail address: xiangyu.wang@curtin.edu.au
E-mail address: jaemlee@pusan.ac.kr
E-mail address: kjung@pusan.ac.kr