非平滑凸最適化に関する並列計算手法
Parallel Computing Method for Nonsmooth Convex Optimization

明治大学理工学部情報科学科　菱沼 和弘, 飯塚 秀明

Kazuhiro HISHINUMA and Hideaki IIDUKA
Department of Computer Science, School of Science and Technology,
Meiji University

Abstract. In this paper, we consider the problem of minimizing the sum of nondifferentiable, convex functions over a closed convex set in a real Hilbert space, which is simple in the sense that the projection onto it can be easily calculated. We present a parallel subgradient method for solving it and the two convergence analyses of the method. One analysis shows that the parallel method with a small constant step size approximates a solution to the problem. The other analysis indicates that the parallel method with a diminishing step size converges to a solution to the problem in the sense of the weak topology of the Hilbert space. Finally, we numerically compare our method with the existing method and state future work on parallel subgradient methods.

1. Introduction

This paper considers the following standard nonsmooth convex minimization problem.

Problem 1.1. Let f_i $(i = 1, 2, \ldots, K)$ be convex, continuous functionals on a real Hilbert space H and let C be a nonempty, closed convex subset of H. Then,

$$\text{minimize } \sum_{i=1}^{K} f_i(x) \text{ subject to } x \in C.$$

A useful algorithm for solving Problem 1.1 is the *incremental subgradient method* [9, 12], and it is defined as follows: for defining P_C as the projection onto C and $\partial f_i(x)$ as the subdifferential of f_i at $x \in H$ $(i = 1, 2, \ldots, K)$, an iteration $(n + 1)$ of the algorithm is

$$\begin{cases} \psi_{0,n} := x_n, \\ \psi_{i,n} := P_C\left(\psi_{i-1,n} - \lambda_n g_{i,n}\right),\ g_{i,n} \in \partial f_i\left(\psi_{i-1,n}\right) \quad (i = 1, 2, \ldots, K), \\ x_{n+1} := \psi_{K,n}. \end{cases} \tag{1.1}$$

Algorithm (1.1) requires us to use P_C each iteration. Hence, we assume that C is simple in the sense that P_C can be easily calculated within a finite number of arithmetic operations [1, p.406], [2, Subchapter 28.3]. Some incremental methods that can be applied when C is not always simple were presented in [4, 6, 7].

Meanwhile, *parallel proximal algorithms* [2, Proposition 27.8], [3, Algorithm 10.27], [10] are also useful for solving Problem 1.1. They use the *proximity operator* of a nondifferentiable, convex f_i which maps every $x \in H$ to the unique minimizer of $f_i + (1/2)\|x - \cdot\|^2$, where $\|\cdot\|$ stands for the norm of H. The parallel gradient algorithms presented in [6, 7] work only when f_i is differentiable and convex, and C is not always simple.

This paper presents a *parallel subgradient method* for solving Problem 1.1. The proposed method does not use any proximity operators, in contrast to the algorithms in [2, Proposition 27.8], [3, Algorithm 10.27], [10]. Next, we present convergence analyses for the two step-size rules: a constant step-size rule and a diminishing step-size rule. We show that the proposed method with a small constant step size approximates a solution to Problem 1.1. We also show that the algorithm with a diminishing step size weakly converges to a solution to Problem 1.1.

This paper is organized as follows. Section 2 gives the mathematical preliminaries. Section 3 presents the parallel algorithm for minimizing the sum of convex functionals over a simple, convex closed constraint set and studies its convergence properties for a constant step size and a diminishing step size. Section 4 provides numerical examples of the algorithm. Section 5 concludes the paper and mentions future work on parallel subgradient methods.

2. Preliminaries

Let H be a real Hilbert space with inner product $\langle \cdot, \cdot \rangle$, and its induced norm $\|\cdot\|$. Let $\mathbb{N}$ denote the set of all positive integers including zero.

2.1. **Subdifferentiability and projection.**

The *subdifferential* [2, Definition 16.1], [11, Section 23], [13, p.132] of $f \colon H \to \mathbb{R}$ is the set-valued operator,

$$\partial f \colon H \to 2^H \colon x \mapsto \{u \in H \colon f(y) \geq f(x) + \langle y - x, u \rangle \ \ (y \in H)\}.$$

Suppose that $f \colon H \to \mathbb{R}$ is continuous and convex with $\mathrm{dom}(f) := \{x \in H \colon f(x) < \infty\} = H$. Then, $\partial f(x) \neq \emptyset$ $(x \in H)$ [2, Proposition 16.14(ii)].

Proposition 2.1. [2, Proposition 16.17] *Let $f \colon H \to \mathbb{R}$ be continuous and convex with $\mathrm{dom}(f) = H$. Then, the following are equivalent:*

(i) *f is bounded on every bounded subset of H.*
(ii) *f is Lipschitz continuous relative to every bounded subset of H.*
(iii) *$\mathrm{dom}(\partial f) := \{x \in H \colon \partial f(x) \neq \emptyset\} = H$ and ∂f maps every bounded subset of H to a bounded set.*

The metric projection [2, Subchapter 4.2, Chapter 28] onto a nonempty, closed convex set C $(\subset H)$ is denoted by P_C. It is defined by $P_C(x) \in C$ and $\|x - P_C(x)\| = \inf_{y \in C} \|x - y\|$ $(x \in H)$. P_C is (firmly) nonexpansive with $\mathrm{Fix}(P_C) := \{x \in H \colon P_C(x) = x\} = C$ [2, Proposition 4.8, (4.8)].

2.2. **Main problem.**

This paper deals with a networked system with K users. Throughout this paper, we assume the following.

Assumption 2.2.

(A1) C $(\subset H)$ is a nonempty, closed convex set, and P_C can be easily calculated;

(A2) $f_i \colon H \to \mathbb{R}$ $(i = 1, 2, \ldots, K)$ is continuous and convex with $\mathrm{dom}(f_i) = \mathrm{dom}(\partial f_i) = H$, and $\bigcup\{\partial f_i(x) \colon x \in B\}$ is bounded for a bounded set B $(\subset H)$;
(A3) User i $(i = 1, 2, \ldots, K)$ can use P_C and ∂f_i;
(A4) User i $(i = 1, 2, \ldots, K)$ can communicate with all users.

The main objective of this paper is to solve the following problem.

Problem 2.3. Under Assumption 2.2, find a minimizer of $\sum_{i=1}^{K} f_i$ over C.

3. Parallel Algorithm

We present a parallel algorithm for solving Problem 2.3.

Algorithm 3.1.
Step 0. All users set $x_0 \in H$ arbitrarily and $\{\lambda_n\} \subset (0, \infty)$.
Step 1. User i $(i = 1, 2, \ldots, K)$ computes $y_{i,n} \in H$ as follows:

$$\begin{cases} g_{i,n} \in \partial f_i \left(x_n\right), \\ y_{i,n} := P_C \left(x_n - \lambda_n g_{i,n}\right). \end{cases}$$

Step 2. User i $(i = 1, 2, \ldots, K)$ shares $y_{i,n}$ in Step 1 with all users and calculates $x_{n+1} \in H$ as follows:

$$x_{n+1} := \frac{1}{K} \sum_{i=1}^{K} y_{i,n}.$$

Step 3. Put $n := n + 1$, and go to Step 1.

Assumption (A2) ensures that $\partial f_i(x_n) \neq \emptyset$ $(i = 1, 2, \ldots, K, n \in \mathbb{N})$ [2, Proposition 16.14(ii)]. Assumption (A3) implies that user i $(i = 1, 2, \ldots, K)$ can compute $y_{i,n}$. Moreover, (A4) guarantees that all users can calculate x_n in Step 2.

The convergence analyses of Algorithm 3.1 depend on the following assumption.

Assumption 3.2. For $i = 1, 2, \ldots, K$, there exists $M_i \in \mathbb{R}$ such that

$$\sup \left\{\|g\| : g \in \partial f_i \left(x_n\right),\ n \in \mathbb{N}\right\} < M_i.$$

Suppose that C is bounded (e.g., C is a closed ball). From $\{y_{i,n}\} \subset C$ $(i = 1, 2, \ldots, K)$, $\{y_{i,n}\}$ $(i = 1, 2, \ldots, K)$ is bounded. Accordingly, $\{x_n\}$ is bounded. Hence, (A2) (see Proposition 2.1 for the equivalent properties of (A2)) ensures that Assumption 3.2 holds. Moreover, since (A1) and (A2) imply that $C \cap \mathrm{dom}(f) = C \neq \emptyset$ and C is bounded, (A2) (the continuity and convexity of f) guarantees that Problem 2.3 has a solution [2, Proposition 11.14].

This paper uses the notation,

$$M := \max \left\{M_i \colon i = 1, 2, \ldots, K\right\},$$

$$f := \sum_{i=1}^{K} f_i,\ X := \left\{x \in C \colon f(x) = \inf_{y \in C} f\left(y\right)\right\}.$$

The following is the convergence analysis of Algorithm 3.1 when the step size is some constant.

Theorem 3.3. *Suppose that Assumption 3.2 holds. Let λ be a positive real number and let $\{x_n\} \subset H$ be the sequence generated by Algorithm 3.1. When $\lambda_n := \lambda$ for all $n \in \mathbb{N}$, the following holds.*

$$\liminf_{n\to\infty} f(x_n) \leq \inf_{x\in C} f(x) + \frac{1}{2}\lambda K M^2.$$

Proof. See [5]. □

The following theorem indicates the weak convergence of Algorithm 3.1 with a diminishing step-size sequence.

Theorem 3.4. *Suppose that Assumption 3.2 holds and $\{x_n\} \subset H$ is the sequence generated by Algorithm 3.1, with $\{\lambda_n\}$ satisfying*

$$\sum_{n=0}^{\infty} \lambda_n = \infty \text{ and } \sum_{n=0}^{\infty} \lambda_n^2 < \infty.$$

Then, if X is nonempty, $\{x_n\}$ converges weakly to some point in X.

Proof. See [5]. □

4. Numerical Examples

We applied the incremental subgradient method (1.1) and Algorithm 3.1 to the following N-dimensional constrained nonsmooth convex optimization problem (Problem 1.1 when $H = \mathbb{R}^N$ and $K = N$).

Problem 4.1. Let $f_i : \mathbb{R}^N \to \mathbb{R}$ $(i = 1, 2, \ldots, N)$ be convex and let C be a nonempty, closed convex subset of $\mathbb{R}^N$. Then,

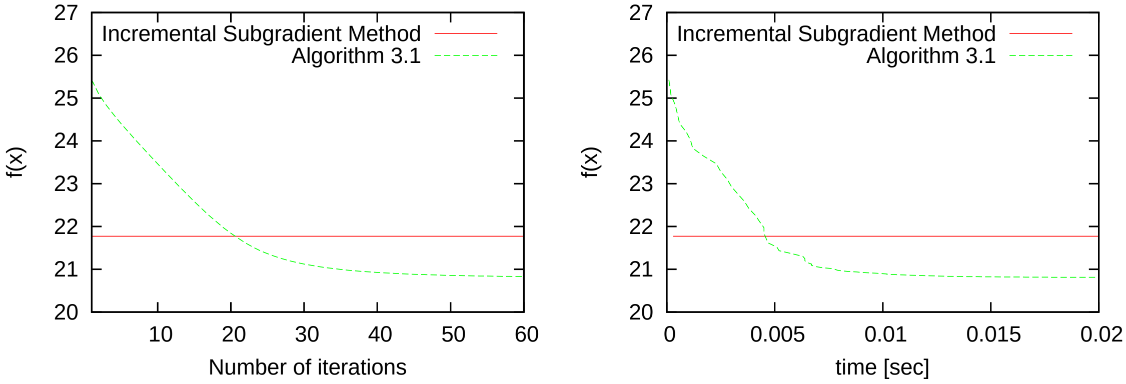
$$\text{minimize } \sum_{i=1}^{N} f_i(x) \text{ subject to } x \in C.$$

In the experiment, we used the PC-Cluster composed of 48 *Fujitsu PRIMERGY RX350 S7* computers at the Ikuta campus of Meiji University. One of those computers has two *Xeon E5-2690* (2.9GHz, 8 cores) CPUs and 32GB memory. We used 64 CPU cores of this cluster; i.e., there were 64 users in the experiment environment that satisfied (A3) and (A4) of the Assumption 2.2. In the implementation of Step 2 in Algorithm 3.1, we used the `MPI_Allreduce` function, which is categorized as an **All-To-All** collective operation in [8, Chapter 5], to compute and share the sum of $y_{i,n}$ with all users. This means that all users contributed to computing x_{n+1} in Algorithm 3.1. This operation does not violate Assumption 2.2. The experimental programs were written in C and compiled by *gcc version 4.4.7* with *Intel(R) MPI Library 4.1*. We used *GNU Scientific Library 1.16* to express and compute vectors.

We set $N := 64$ and $C := \{x \in \mathbb{R}^N : \|x\| \leq 1\}$ in Problem 4.1. For all $i = 1, 2, \ldots, N$, we prepared random numbers $a_i \in (0, 1)$ and $b_i \in (-1, 1)$ and gave a_i and b_i to user i in advance. The objective function of user i was defined for all $x \in \mathbb{R}^N$ by $f_i(x) := |a_i \langle x, e_i \rangle + b_i|$, where e_i $(i = 1, 2, \ldots, N)$ stands for the natural base of $\mathbb{R}^N$.

In the experiment, we set $\lambda_n := 1$ for the constant step-size rule and $\lambda_n := 1/(n+1)$ for the diminishing step-size rule. We performed 100 samplings, each starting from the different random initial points in $[0, 1)^N$.

Figure 1 shows the behaviors of $f(x) := \sum_{i=1}^N f_i(x)$ for the incremental subgradient method (1.1) and Algorithm 3.1 with a constant step size. The y-axis in Figures 1(a) and 1(b) represent the value of $f(x)$. The x-axis in Figure 1(a) represents the number of iterations and the x-axis in Figure 1(b) represents the elapsed time. The results show that Algorithm 3.1 minimizes the value of $f(x)$ more than the incremental subgradient method does (1.1).



(a) Evaluation by the number of iterations

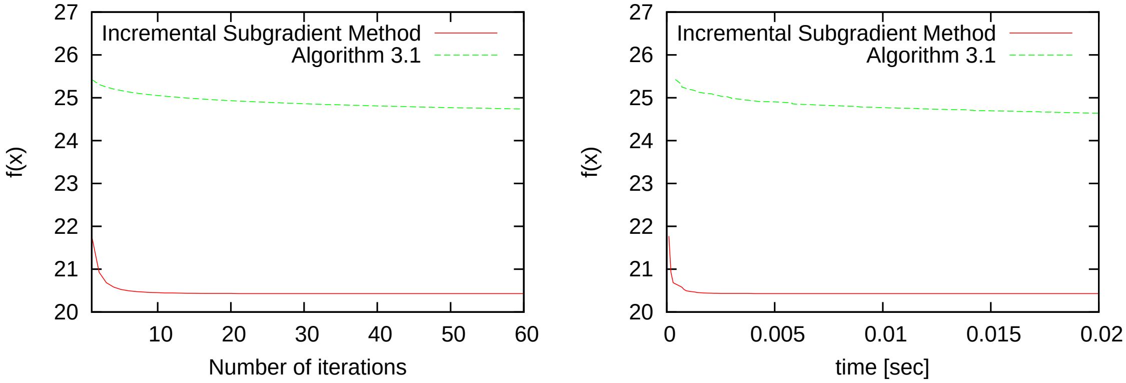
(b) Evaluation by the elapsed time

FIGURE 1. Behavior of $f(x)$ for the incremental subgradient method and Algorithm 3.1 with constant step size

Figure 2 shows the behaviors of $f(x)$ for the incremental subgradient method (1.1) and Algorithm 3.1 with the diminishing step size. The y-axis in Figures 2(a) and 2(b) represent the value of $f(x)$. The x-axis in Figure 2(a) represents the number of iterations, and the x-axis in Figure 2(b) represents the elapsed time. The results show that Algorithm 3.1 converges slower than the incremental subgradient method. However, it shows that Algorithm 3.1 with a constant step size behaves roughly to the same as the incremental subgradient method with the diminishing step size. This implies that, if it is difficult to share the diminishing step size with all users, Algorithm 3.1 can be used as an effective approximation algorithm of the incremental subgradient method.

5. Conclusion and Future Work

This paper discussed the problem of minimizing the sum of nondifferentiable, convex functions over a simple convex closed constraint set of a real Hilbert space. It presented a parallel



(a) Evaluation by the number of iterations

(b) Evaluation by the elapsed time

Figure 2. Behavior of $f(x)$ with diminishing step size

algorithm for solving the problem. We studied its convergence properties for a constant step size and a diminishing step size. We showed that the algorithm with a constant step size approximates a solution to the problem, while the algorithm with a diminishing step size weakly converges to a solution to the problem. Finally, we numerically compared the algorithm with the existing algorithm and showed that, when the step size is constant, the algorithm performs better than the existing algorithm.

The numerical comparisons also indicated that, when the step size is diminishing, the existing algorithm converges to a solution faster than our algorithm. Therefore, in the future, we should consider developing parallel optimization algorithms which perform better than the existing algorithm even when the step sizes are diminishing.

Acknowledgments

The authors would like to thank Professor Wataru Takahashi for giving us a chance to discuss Problem 2.3 in The International Workshop on Nonlinear Analysis and Convex Analysis held at the Research Institute for Mathematical Sciences, Kyoto University on the 19th of August in 2014.

The authors would like to congratulate Professor Wataru Takahashi on his 70th birthday.

References

[1] Bauschke, H.H., Borwein, J.M.: On projection algorithms for solving convex feasibility problems. SIAM Review **38**(3), 367–426 (1996)

[2] Bauschke, H.H., Combettes, P.L.: Convex Analysis and Monotone Operator Theory in Hilbert Spaces. Springer (2011)

[3] Combettes, P.L., Pesquet, J.C.: Proximal splitting methods in signal processing. 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. 185–212. Springer (2011)

[4] Helou Neto, E.S., De Pierro, A.R.: Incremental subgradients for constrained convex optimization: A unified framework and new methods. SIAM Journal on Optimization **20**, 1547–1572 (2009)

[5] Hishinuma, K., Iiduka, H.: Parallel subgradient method for nonsmooth convex optimization with a simple constraint. (submitted)

[6] Iiduka, H.: Fixed point optimization algorithms for distributed optimization in networked systems. SIAM Journal on Optimization **23**, 1–26 (2013)

[7] Iiduka, H., Hishinuma, K.: Acceleration method combining broadcast and incremental distributed optimization algorithms. SIAM Journal on Optimization **24**(4), 1840–1863 (2014)

[8] Message Passing Interface Forum: MPI: A Message-Passing Interface Standard, Version 3.0. High Performance Computing Center Stuttgart (2012)

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

[10] Pesquet, J.C., Pustelnik, N.: A parallel inertial proximal optimization method. Pacific Journal of Optimization **8**, 273–306 (2012)

[11] Rockafellar, R.T.: Convex Analysis. Princeton University Press (1970)

[12] Solodov, M.V., Zavriev, S.K.: Error stability properties of generalized gradient-type algorithms. Journal of Optimization Theory and Applications **98**, 663–680 (1998)

[13] Takahashi, W.: Nonlinear Functional Analysis. Yokohama Publishers (2000)

(K. Hishinuma, H. Iiduka) Department of Computer Science, Meiji University, 1-1-1 Higashimita, Tama-ku, Kawasaki-shi, Kanagawa 214-8571, Japan

E-mail address, K. Hishinuma: `kaz@cs.meiji.ac.jp`

E-mail address, H. Iiduka: `iiduka@cs.meiji.ac.jp`