Hindawi Publishing Corporation
Journal of Applied Mathematics
Volume 2012, Article ID 139609, 14 pages
doi:10.1155/2012/139609

Research Article

New Nonsmooth Equations-Based Algorithms for ℓ_1-Norm Minimization and Applications

Lei Wu[1] and Zhe Sun[2]

[1] *College of Mathematics and Econometrics, Hunan University, Changsha 410082, China*
[2] *College of Mathematics and Information Science, Jiangxi Normal University, Nanchang 330022, China*

Correspondence should be addressed to Zhe Sun, snzma@126.com

Received 18 September 2012; Revised 8 December 2012; Accepted 9 December 2012

Academic Editor: Changbum Chun

Recently, Xiao et al. proposed a nonsmooth equations-based method to solve the ℓ_1-norm minimization problem (2011). The advantage of this method is its simplicity and lower storage. In this paper, based on new nonsmooth equations reformulation, we investigate new nonsmooth equations-based algorithms for solving ℓ_1-norm minimization problems. Under mild conditions, we show that the proposed algorithms are globally convergent. The preliminary numerical results demonstrate the effectiveness of the proposed algorithms.

1. Introduction

We consider the ℓ_1-norm minimization problem

$$\min_x f(x) \triangleq \frac{1}{2}\|Ax-b\|^2+\rho\|x\|_1, \tag{1.1}$$

where $x \in \mathbb{R}^n$, $b \in \mathbb{R}^m$, $A \in \mathbb{R}^{m\times n}$, and ρ is a nonnegative parameter. Throughout the paper, we use $\|v\| = \sqrt{\sum_{i=1}^n |v_i|^2}$ and $\|v\|_1 = \sum_i |v_i|$ to denote the Euclidean norm and the ℓ_1-norm of vector $v \in R^n$, respectively. Problem (1.1) has many important practical applications, particularly in compressed sensing (abbreviated as CS) [1] and image restoration [2]. It can also be viewed as a regularization technique to overcome the ill-conditioned, or even singular, nature of matrix A, when trying to infer x from noiseless observations $b = Ax$ or from noisy observations $b = Ax + \xi$, where ξ is the white Gaussian noise of variance σ^2 [3–5].

The convex optimization problem (1.1) can be cast as a second-order cone programming problem and thus could be solved via interior point methods. However, in many applications, the problem is not only large scale but also involves dense matrix data, which often precludes the use and potential advantage of sophisticated interior point methods. This motivated the search of simpler first-order algorithms for solving (1.1), where the dominant computational effort is a relatively cheap matrix-vector multiplication involving A and A^T. In the past few years, several first-order algorithms have been proposed. One of the most popular algorithms falls into the iterative shrinkage/thresholding (IST) class [6, 7]. It was first designed for wavelet-based image deconvolution problems [8] and analyzed subsequently by many authors, see, for example, [9–11]. Figueiredo et al. [12] studied the gradient projection and Barzilai-Borwein method [13] (denoted by GPSR-BB) for solving (1.1). They reformulated problem (1.1) as a box-constrained quadratic program and solved it by a gradient projection and Barzilai-Borwein method. Wright et al. [14] presented sparse reconstruction algorithm (denoted by SPARSA) to solve (1.1). Yun and Toh [15] proposed a block coordinate gradient descent algorithm for solving (1.1). Yang and Zhang [16] investigated alternating direction algorithms for solving (1.1).

Quite recently, Xiao et al. [17] developed a nonsmooth equations-based algorithm (called SGCS) for solving ℓ_1-norm minimization problems in CS. They reformulated the box-constrained quadratic program obtained by Figueiredo et al. [12] into a system of nonsmooth equations and then applied the spectral gradient projection method [18] to solving the nonsmooth equation. The main advantage of the SGCS is its simplicity and lower storage. The difference between the above algorithms and SGCS is that SGCS did not use line search to decrease the value of objective function at each iteration and instead used a projection step to accelerate the iterative process. However, each projection step in SGCS requires two matrix-vector multiplication involving A or A^T, which means that each iteration requires matrix-vector multiplication involving A or A^T four times, while each iteration in GPSR-BB and IST is only two times. This may bring in more computational complexity. In addition, the dimension of the system of nonsmooth equations is $2n$, which is twice of the original problems. These drawbacks motivate us to study new nonsmooth equations-based algorithms for the ℓ_1-norm minimization problem.

In this paper, we first reformulate problem (1.1) into a system of nonsmooth equations. This system is Lipschitz continuous and monotone and many effective algorithms (see, e.g., [18–22]) can be used to solve it. We then apply spectral gradient projection (denoted by SGP) method [18] to solve the resulting system. Similar to SGCS, each iteration in SGP requires matrix-vector multiplication involving A or A^T four times. In order to reduce the computational complexity, we also propose a modified SGP (denoted by MSGP) method to solve the resulting system. Under mild conditions, the global convergence of the proposed algorithms will be ensured.

The remainder of the paper is organized as follows. In Section 2, we first review some existing results of nonsmooth analysis and then derive an equivalent system of nonsmooth equations to problem (1.1). We verify some nice properties of the resulting system in this section. In Section 3, we propose the algorithms and establish their global convergence. In Section 4, we apply the proposed algorithms to some practical problems arising from compressed sensing and image restoration and compare their performance with that of SGCS, SPARSA, and GPSR-BB.

Throughout the paper, we use $\langle \cdot\, , \cdot \rangle$ to denote the inner product of two vectors in R^n.

2. Preliminaries

By nonsmooth analysis, a necessary condition for a vector $x \in \mathbb{R}^n$ to be a local minima of nonsmooth function $f : \mathbb{R}^n \to \mathbb{R}$ is that

$$(0, \dots, 0)^T \in \partial f(x), \tag{2.1}$$

where $\partial f(x)$ denotes the subdifferential of f at x [23]. If f is convex, then (2.1) is also sufficient for x to be a solution of (1.1). The subdifferential of the absolute value function $|t|$ is given by the signum function $\operatorname{sign}(t)$, that is

$$\partial|t| = \operatorname{sign}(t) := \begin{cases} \{1\}, & t > 0, \\ [-1, 1], & t = 0, \\ \{-1\}, & t < 0. \end{cases} \tag{2.2}$$

For problem (1.1), the optimality conditions therefore translate to

$$\begin{gathered} \nabla_i f(x) + \rho \operatorname{sign}(x_i) = 0, \quad |x_i| > 0, \\ \left|\nabla_i f(x)\right| \le \rho, \quad x_i = 0, \end{gathered} \tag{2.3}$$

where $\nabla_i f(x) = \partial f(x)/\partial x_i$, $i = 1, \dots, n$. It is clear that the function defined by (1.1) is convex. Therefore a point $x^* \in R^n$ is a solution of problem (1.1) if and only if it satisfies

$$\begin{aligned} \nabla_i f(x^*) + \rho = 0, &\quad \text{if } x_i^* > 0, \\ \nabla_i f(x^*) - \rho = 0, &\quad \text{if } x_i^* < 0, \\ -\rho \le \nabla_i f(x^*) \le \rho, &\quad \text{if } x_i^* = 0. \end{aligned} \tag{2.4}$$

Formally, we call the above conditions the *optimality conditions* for problem (1.1).

For any given $\tau > 0$, we define a mapping $H^\tau = (H_1^\tau, H_2^\tau, \dots, H_n^\tau)^T : \mathbb{R}^n \to \mathbb{R}^n$ by

$$H_i^\tau(x) \triangleq \max\left\{\tau\left(\nabla_i f(x) - \rho\right), \min\left\{x_i, \tau\left(\nabla_i f(x) + \rho\right)\right\}\right\}. \tag{2.5}$$

Then H^τ is a continuous mapping and is closely related to problem (1.1). It is generally not differentiable in the sense of Fréchet derivative but semismooth in the sense of Qi and Sun [24]. The following proposition shows that the ℓ_1-norm minimization problem (1.1) is equivalent to a nonsmooth equation. It can be easily obtained by the use of the optimality conditions and the convexity of the function f defined by (1.1).

Proposition 2.1. *Let $\tau > 0$ be any given constant. A point $x^* \in \mathbb{R}^n$ is a solution of problem* (1.1) *if and only if it satisfies*

$$H^\tau(x^*) = 0. \tag{2.6}$$

The above proposition has reformulated problem (1.1) as a system of nonsmooth equations. Compared with the nonsmooth equation reformulation in [17], the dimension of (2.6) is only half of the dimension of the equation in [17].

Given $a, b, c, d \in \mathbb{R}$. It is easy to verify that (see, e.g. [25])

$$\begin{aligned} \min\{a,b\} - \min\{c,d\} &= (1-s)(a-c) + s(b-d), \\ \max\{a,b\} - \max\{c,d\} &= (1-t)(a-c) + t(b-d) \end{aligned} \tag{2.7}$$

with

$$\begin{aligned} s &= \begin{cases} 0, & a \le b,\ c \le d; \\ 1, & a > b,\ c > d; \\ \dfrac{\min\{a,b\} - \min\{c,d\} + c - a}{b - d + c - a}, & \text{otherwise}, \end{cases} \\ t &= \begin{cases} 0, & a \ge b,\ c \ge d; \\ 1, & a < b,\ c < d; \\ \dfrac{\max\{a,b\} - \max\{c,d\} + c - a}{b - d + c - a}, & \text{otherwise}. \end{cases} \end{aligned} \tag{2.8}$$

It is clear that $0 \le s, t \le 1$. By (2.5), we have for any $x, y \in \mathbb{R}^n$, it holds that

$$\begin{aligned} &H_i^{\tau}(x) - H_i^{\tau}(y) \\ &\quad = \max\{\tau(\nabla_i f(x) - \rho), \min\{x_i, \tau(\nabla_i f(x) + \rho)\}\} \\ &\qquad - \max\{\tau(\nabla_i f(y) - \rho), \min\{y_i, \tau(\nabla_i f(y) + \rho)\}\} \\ &\quad = \tau(1 - t_i)(\nabla_i f(x) - \nabla_i f(y)) \\ &\qquad + t_i(\min\{x_i, \tau(\nabla_i f(x) + \rho)\} - \min\{y_i, \tau(\nabla_i f(y) + \rho)\}) \\ &\quad = \tau(1 - t_i)(\nabla_i f(x) - \nabla_i f(y)) \\ &\qquad + t_i((1 - s_i)(x_i - y_i) + \tau s_i(\nabla_i f(x) - \nabla_i f(y))) \\ &\quad = t_i(1 - s_i)(x_i - y_i) \\ &\qquad + \tau(1 - t_i + t_i s_i)(\nabla_i f(x) - \nabla_i f(y)), \end{aligned} \tag{2.9}$$

where $0 \le s_i, t_i \le 1$. Define two diagonal matrixes S and T by

$$S = \operatorname{diag}\{s_1, s_2, \ldots, s_n\}, \qquad T = \operatorname{diag}\{t_1, t_2, \ldots, t_n\}. \tag{2.10}$$

Then we obtain

$$H^{\tau}(x) - H^{\tau}(y) = T(I - S)(x - y) + \tau(I - T + TS)(\nabla f(x) - \nabla f(y)). \tag{2.11}$$

Since $\nabla f(x) = A^T(Ax - b)$, we get

$$H^\tau(x) - H^\tau(y) = \Big(T(I-S) + \tau(I - T + TS)A^T A\Big)(x-y). \tag{2.12}$$

The next proposition shows the Lipschitz continuity of H^τ defined by (2.5).

Proposition 2.2. *For each $\tau > 0$, there exists a positive constant $L(\tau)$ such that*

$$\|H^\tau(x) - H^\tau(y)\| \le L(\tau)\|x-y\|, \quad \forall x, y \in \mathbb{R}^n. \tag{2.13}$$

Proof. By (2.10) and (2.12), we have

$$\|H^\tau(x) - H^\tau(y)\| \le \left\|T(I-S) + \tau(I-T+TS)A^T A\right\| \|x-y\| \le \left(1 + \tau\left\|A^T A\right\|\right)\|x-y\|. \tag{2.14}$$

Let $L(\tau) \triangleq 1 + \tau\|A^T A\|$. Then (2.13) holds. The proof is complete. □

The following proposition shows another good property of the system of nonsmooth equations (2.6).

Proposition 2.3. *There exists a constant $\tau^* > 0$ such that for any $0 < \tau \le \tau^*$, the mapping $H^\tau : \mathbb{R}^n \to \mathbb{R}^n$ is monotone, that is*

$$\langle H^\tau(x) - H^\tau(y), x - y\rangle \ge 0, \quad \forall x, y \in \mathbb{R}^n. \tag{2.15}$$

Proof. Let D_{ii} be the ith diagonal element of $A^T A$. It is clear that $D_{ii} > 0$, $i = 1, \ldots, n$. Set $\tau^* \triangleq \min_i\{1/D_{ii}\}$. Note that $A^T A$ is symmetric and positive semidefinite. Consequently, for any $\tau \in (0, \tau^*]$, matrix $T(I-S) + \tau(I-T+TS)A^T A$ is also positive semidefinite. Therefore, it follows from (2.12) that

$$\langle H^\tau(x) - H^\tau(y), x - y\rangle \ge 0. \tag{2.16}$$

This completes the proof. □

3. Algorithms and Their Convergence

In this section, we describe the proposed algorithms in detail and establish their convergence. Let $\tau > 0$ be given. For simplicity, we omit τ and abbreviate $H^\tau(\cdot)$ as $H(\cdot)$.

Algorithm 3.1 (spectral gradient projection method (abbreviated as SGP)). Given initial point $x_0 \in \mathbb{R}^n$ and constants $\theta_0 = 1$, $r > 0$, $\nu \ge 0$, $\sigma > 0$, $\gamma \in (0,1)$. Set $k := 0$.

Step 1. Compute d_k by

$$d_k = -\theta_k H(x_k), \tag{3.1}$$

where for each $k \geq 1$, θ_k is defined by

$$\theta_k = \frac{s_{k-1}^T s_{k-1}}{y_{k-1}^T s_{k-1}} \tag{3.2}$$

with $s_{k-1} = x_k - x_{k-1}$ and $y_{k-1} = H(x_k) - H(x_{k-1}) + r\|H(x_k)\|^\nu s_{k-1}$. Stop if $d_k = 0$.

Step 2. Determine steplength $\alpha_k = \gamma^{m_k}$ with m_k being the smallest nonnegative integer m such that

$$-\langle H(x_k + \gamma^m d_k), d_k \rangle \geq \sigma \gamma^m \|H(x_k + \gamma^m d_k)\| \|d_k\|. \tag{3.3}$$

Set $z_k := x_k + \alpha_k d_k$. Stop if $\|H(z_k)\| = 0$.

Step 3. Compute

$$x_{k+1} = x_k - \frac{\langle H(z_k), x_k - z_k \rangle}{\|H(z_k)\|^2} H(z_k). \tag{3.4}$$

Set $k := k + 1$, and go to Step 1.

Remark 3.2. (i) The idea of the above algorithm comes from [18]. The major difference between Algorithm 3.1 and the method in [18] lies in the definition of y_{k-1}. The choice of y_{k-1} in Step 1 follows from the modified BFGS method [26]. The purpose of the term $r\|H(x_k)\|^\nu s_{k-1}$ is to make y_{k-1} be closer to $H(x_k) - H(x_{k-1})$ as x_k tends to a solution of (2.6).

(ii) Step 3 is called the projection step. It is originated in [20]. The advantage of the projection step is to make x_{k+1} closer to the solution set of (2.6) than x_k. We refer to [20] for details.

(iii) Since $-\langle H(x_k), d_k \rangle = \|H(x_k)\| \|d_k\|$, by the continuity of H, it is easy to see that inequality (3.3) holds for all m sufficiently large. Therefore Step 2 is well defined and so is Algorithm 3.1.

The following lemma comes from [20].

Lemma 3.3. *Let $H : \mathbb{R}^n \rightarrow \mathbb{R}^n$ be monotone and $x, y \in \mathbb{R}^n$ satisfy $\langle H(y), x - y \rangle > 0$. Let*

$$x^+ = x - \frac{\langle H(y), x - y \rangle}{\|H(y)\|^2} H(y). \tag{3.5}$$

Then for any $x^* \in \mathbb{R}^n$ satisfying $H(x^*) = 0$, it holds that

$$\|x^+ - x^*\|^2 \leq \|x - x^*\|^2 - \|x^+ - x\|^2. \tag{3.6}$$

The following theorem establishes the global convergence for Algorithm 3.1.

Theorem 3.4. *Let $\{x_k\}$ be generated by Algorithm 3.1 and x^* a solution of* (2.6)*. Then one has*

$$\|x_{k+1} - x^*\|^2 \leq \|x_k - x^*\|^2 - \|x_{k+1} - x_k\|^2. \tag{3.7}$$

In particular, $\{x_k\}$ is bounded. Furthermore, it holds that either $\{x_k\}$ is finite and the last iterate is a solution of the system of nonsmooth equations (2.6)*, or the sequence is infinite and $\lim_{k\to\infty}\|x_{k+1} - x_k\| = 0$. Moreover, $\{x_k\}$ converges to some solution of* (2.6)*.*

Proof. The proof is similar to that in [18]. We omit it here. □

Remark 3.5. The computational complexity of each of SGP's steps is clear. In large-scale problems, most of the work is matrix-vector multiplication involving A and A^T. Steps 1 and 2 of SGP require matrix-vector multiplication involving A or A^T two times each, while each iteration in GPSR-BB involves matrix-vector multiplication only two times. This may bring in more computational complexity. Therefore, we give a modification of SGP. The modified algorithm, which will be called MSGP in the rest of the paper, coincides with SGP except at Step 3, whose description is given below.

Algorithm 3.6 (modified spectral gradient projection method (abbreviated as MSGP)). Given initial point $x_0 \in \mathbb{R}^n$ and constants $\theta_0 = 1$, $r > 0$, $\sigma > 0$, $\gamma \in (0,1)$ a positive integer M. Set $k := 0$.

Step 3. Let $m = k/M$. If m is a positive integer, compute

$$x_{k+1} = x_k - \frac{\langle H(z_k), x_k - z_k\rangle}{\|H(z_k)\|^2} H(z_k); \tag{3.8}$$

otherwise, let $x_{k+1} = z_k$. Set $k := k+1$, and go to Step 1.

Lemma 3.7. *Assume that $\{x_k\}$ is a sequence generated by Algorithm 3.6 and $x^* \in \mathbb{R}^n$ satisfies $H(x^*) = 0$. Let $\lambda_{\max}(A^TA)$ be the maximum eigenvalue of A^TA and $\tau \in (0, 1/\lambda_{\max}(A^TA)]$. Then it holds that*

$$\|x_{k+1} - x^*\| \leq \|x_k - x^*\|, \quad k = 0, 1, 2, \ldots. \tag{3.9}$$

Proof. Let x_{k+1} be generated by (3.8). It follows from Lemma 3.3 that (3.9) holds. In the following, we assume that $x_{k+1} = z_k$. Then, we obtain

$$\begin{aligned}
\|x_{k+1} - x^*\| &= \|x_k + \alpha_k d_k - x^*\| \\
&= \|x_k - \alpha_k\theta_k H(x_k) - x^* + \alpha_k\theta_k H(x^*)\| \\
&= \|(x_k - x^*) - \alpha_k\theta_k[H(x_k) - H(x^*)]\|.
\end{aligned} \tag{3.10}$$

This together with (2.12) implies that

$$\begin{aligned}\|x_{k+1}-x^*\| &= \left\|x_k - x^* - \alpha_k\theta_k\left[\left(T_k(I-S_k)+\tau(I-T_k+T_kS_k)A^TA\right)\right](x_k-x^*)\right\| \\ &= \left\|(1-\alpha_k\theta_k)(x_k-x^*)+\alpha_k\theta_k(I-T_k+T_kS_k)\left(I-\tau A^TA\right)(x_k-x^*)\right\| \\ &\le (1-\alpha_k\theta_k)\|x_k-x^*\|+\alpha_k\theta_k\left\|I-\tau A^TA\right\|\|x_k-x^*\|.\end{aligned}\tag{3.11}$$

Let $\tau \in (0, 1/\lambda_{\max}(A^TA)]$. Then we get

$$\|x_{k+1}-x^*\| \le \|x_k-x^*\|. \tag{3.12}$$

This completes the proof. □

Now we establish a global convergence theorem for Algorithm 3.6.

Theorem 3.8. *Let $\lambda_{\max}(A^TA)$ be the maximum eigenvalue of A^TA and $\tau \in (0, 1/\lambda_{\max}(A^TA)]$. Assume that $\{x_k\}$ is generated by Algorithm 3.6 and x^* is a solution of* (2.6). *Then one has*

$$\|x_{k+1}-x^*\| \le \|x_k-x^*\|, \quad k=0,1,2,\ldots. \tag{3.13}$$

In particular, $\{x_k\}$ is bounded. Furthermore, it holds that either $\{x_k\}$ is finite and the last iterate is a solution of the system of nonsmooth equations (2.6), *or the sequence is infinite and $\lim_{k\to\infty}\|x_{k+1}-x_k\| = 0$. Moreover, $\{x_k\}$ converges to some solution of* (2.6).

Proof. We first note that if the algorithm terminates at some iteration k, then we have $d_k = 0$ or $\|H(z_k)\| = 0$. By the definition of θ_k, we have $H(x_k) = 0$ if $d_k = 0$. This shows that either x_k or z_k is a solution of (2.6).

Suppose that $d_k \neq 0$ and $\|H(z_k)\| \neq 0$ for all k. Then an infinite sequence $\{x_k\}$ is generated. It follows from (3.3) that

$$\langle H(z_k), x_k - z_k\rangle = -\alpha_k\langle H(z_k), d_k\rangle \ge \sigma\alpha_k^2\|H(z_k)\|\|d_k\| > 0. \tag{3.14}$$

Let x^* be an arbitrary solution of (2.6). By Lemmas 3.7 and 3.3, we obtain

$$\begin{gathered}\|x_{k+1}-x^*\| \le \|x_k-x^*\|, \\ \|x_{mM+1}-x^*\|^2 \le \|x_{mM}-x^*\|^2 - \|x_{mM+1}-x_{mM}\|^2,\end{gathered}\tag{3.15}$$

where m is a nonnegative integer. In particular, the sequence $\{\|x_k-x^*\|\}$ is nonincreasing and hence convergent. Moreover, the sequence $\{x_k\}$ is bounded, and

$$\lim_{m\to\infty}\|x_{mM+1}-x_{mM}\| = 0. \tag{3.16}$$

Following from (3.8) and (3.14), we have

$$\|x_{mM+1} - x_{mM}\| = \frac{\langle H(z_{mM}), x_{mM} - z_{mM} \rangle}{\|H(z_{mM})\|} \geq \sigma \alpha_{mM}^2 \|d_{mM}\|. \tag{3.17}$$

This together with (3.16) yields

$$\lim_{m \to \infty} \alpha_{mM} \|d_{mM}\| = 0. \tag{3.18}$$

Now we consider the following two possible cases:

(i) $\liminf_{m \to \infty} \|H(x_{mM})\| = 0$;

(ii) $\liminf_{m \to \infty} \|H(x_{mM})\| = \epsilon > 0$.

If (i) holds, then by the continuity of H and the boundedness of $\{x_{mM}\}$, it is clear that the sequence $\{x_{mM}\}$ has some accumulation point x^* such that $H(x^*) = 0$. Since the sequence $\{\|x_k - x^*\|\}$ converges, it must hold that $\{x_k\}$ converges to x^*.

If (ii) holds, then by the boundedness of $\{x_{mM}\}$ and the continuity of H, there exist a positive constant C and a positive integer m_0 such that

$$\frac{1}{2}\epsilon \leq \|H(x_{mM})\| \leq C, \quad \forall m \geq m_0. \tag{3.19}$$

On the other hand, from (3.2) and the definitions of s_{k-1} and y_{k-1}, we have

$$\theta_{mM} = \frac{s_{mM-1}^T s_{mM-1}}{y_{mM-1}^T s_{mM-1}} = \frac{s_{mM-1}^T s_{mM-1}}{\langle H(x_{mM}) - H(x_{mM-1}), x_{mM} - x_{mM-1} \rangle + r\|H(x_{mM})\|^{\nu} s_{mM-1}^T s_{mM-1}}, \tag{3.20}$$

which together with (3.19) and Propositions 2.2 and 2.3 implies

$$\frac{1}{L + rC^{\nu}} \leq \theta_{mM} \leq \frac{2^{\nu}}{r\epsilon^{\nu}}, \quad \forall m \geq m_0. \tag{3.21}$$

Consequently, we obtain from (3.1), (3.19), and (3.21)

$$\|d_{mM}\| = \theta_{mM} \|H(x_{mM})\| \geq \frac{\epsilon}{2(L + rC^{\nu})}, \quad \|d_{mM}\| \leq \frac{2^{\nu} C}{r\epsilon^{\nu}}, \quad \forall m \geq m_0. \tag{3.22}$$

Therefore, it follows from (3.18) that $\lim_{m \to \infty} \alpha_{mM} = 0$. By the line search rule, we have for all m sufficiently large, $m_k - 1$ will not satisfy (3.3). This means

$$-\left\langle H\left(x_{mM} + \gamma^{m_k - 1} d_{mM}\right), d_{mM} \right\rangle < \sigma \gamma^{m_k - 1} \left\| H\left(x_{mM} + \gamma^{m_k - 1} d_{mM}\right) \right\| \|d_{mM}\|. \tag{3.23}$$

Since $\{x_{mM}\}$ and $\{d_{mM}\}$ are bounded, we can choose subsequences of $\{x_{mM}\}$ and $\{d_{mM}\}$ converging to x^{**} and d^{**}, respectively. Taking the limit in (3.23) for the subsequence, we obtain

$$-\langle H(x^{**}), d^{**}\rangle \leq 0. \tag{3.24}$$

However, it is not difficult to deduce from (3.1), (3.19), and (3.21) that

$$-\langle H(x^{**}), d^{**}\rangle > 0. \tag{3.25}$$

This yields a contradiction. Consequently, $\liminf_{m\to\infty}\|H(x_{mM})\| = \epsilon > 0$ is not possible. The proof is then complete. □

4. Applications to Compressed Sensing and Image Restoration

In this section, we apply the proposed algorithms, that is, SGP and MSGP, to solve some practical problems arising from the compressed sensing and image restoration. We will compare the proposed algorithms with SGCS, SPARSA, and GPSR-BB. The system of nonsmooth equations in SGCS is

$$F(z) \triangleq \min\{z, \tau(Hz + c)\} = 0, \tag{4.1}$$

where z, c, H are defined as those in [17]. The test problems are associated with applications in the areas of compressed sensing and image restoration. All experiments were carried out on a Lenovo PC (2.53 GHz, 2.00 GB of RAM) using Matlab 7.8. The parameters in SGCS are specified as follows:

$$\tau = 7, \qquad \beta = 1, \qquad \gamma = 1.2, \qquad \xi = 10^{-4}, \qquad \rho = 0.1. \tag{4.2}$$

The parameters in SGP and MSGP are specified as follows:

$$\tau = 7, \qquad \sigma = 1, \qquad r = 0.8, \qquad \gamma = 0.5, \qquad M = 10. \tag{4.3}$$

Throughout the experiments, we choose the initial iterate to be $x^0 = 0$.

In our first experiment, we consider a typical CS scenario, where the goal is to reconstruct a length-n sparse signal (in the canonical basis) from m observations, where $m < n$. The $m \times n$ matrix A is obtained by first filling it with independent samples of the standard Gaussian distribution and then orthonormalizing the rows. Due to the storage limitations of PC, we test a small size signal with $m = 1024$, $n = 4096$. The observed vector is $b = Ax_{\text{orig}} + \xi$, where ξ is Gaussian white noise with variance $\sigma^2 = 10^{-4}$ and x_{orig} is the original signal with 50 randomly placed ± 1 spikes and with zeros in the remaining elements. The regularization parameter is chosen as $\rho = 0.05\|A^T b\|_\infty$. We compare the performance of SGP and MSGP with that of SGCS, SPARSA, and GPSR-BB by solving the problem and choose

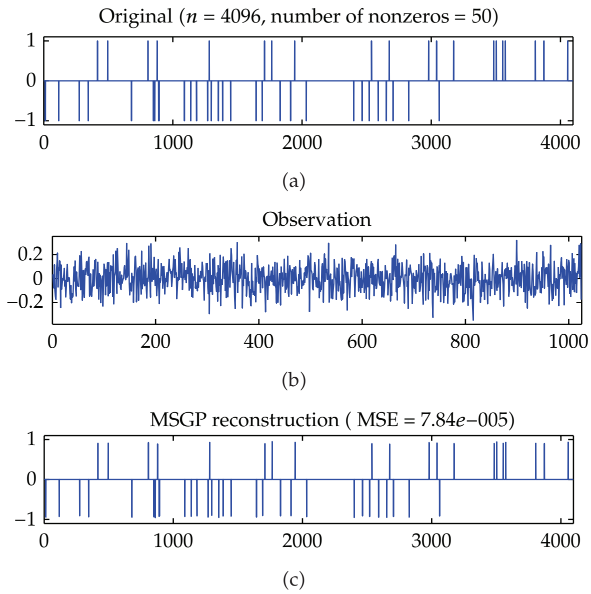


Figure 1: From top to bottom: original signal, observation, and reconstruction obtained by MSGP.

$\nu = 1$ in SGP and MSGP algorithms. We measure the quality of restoration by means of mean squared error (MSE) to the original signal x_{orig} defined by

$$\mathrm{MSE} = \frac{1}{n}\left\|x - x_{\mathrm{orig}}\right\|^2, \tag{4.4}$$

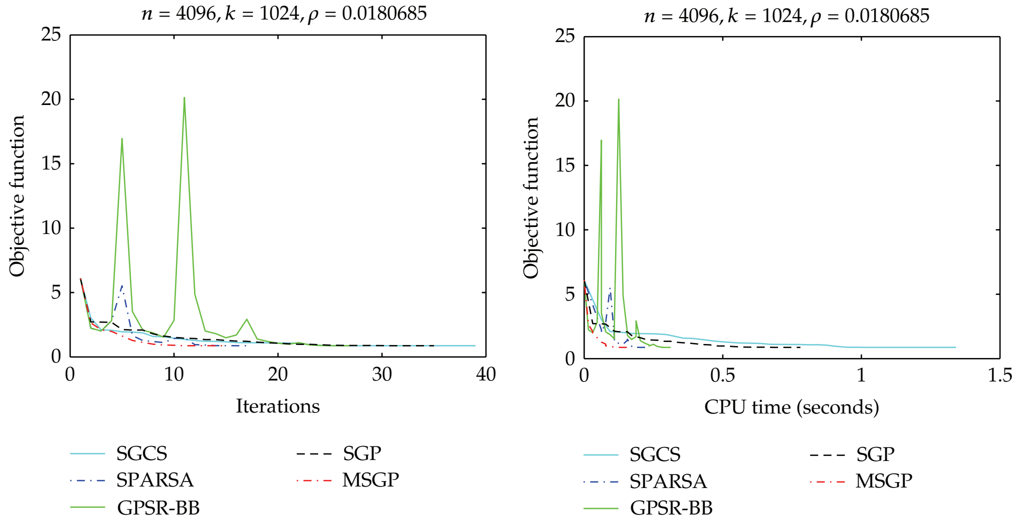
where x is the restored signal. To perform this comparison, we first run the SGCS algorithm and stop the algorithm if the following inequality is satisfied:

$$\frac{\left\|x^{k+1} - x^k\right\|}{\left\|x^k\right\|} < 10^{-5}, \tag{4.5}$$

and then run each of the other algorithms until each reaches the same value of the objective function reached by SGCS.

The original signal and the estimation obtained by solving (1.1) using the MSGP method are shown in Figure 1. We can see from Figure 1 that MSGP does an excellent job at locating the spikes with respect to the original signal. In Figure 2, we plot the evolution of the objective function versus iteration number and CPU time, for these algorithms. It is readily to see that MSGP worked faster than other algorithms.

In the second experiment, we test MSGP for three image restoration problems based on the images as House, Cameraman, and Barbara. House and Cameraman images are of size 256×256 and the other is of size 512×512. All the pixels are contaminated by Gaussian



(a) (b)

Figure 2: The objective function plotted against iteration number and CPU time for SGCS, SPARSA, GPSR-BB, SGP, and MSGP.

noise with the standard deviation of 0.05 with blurring. The blurring function is chosen to be a two-dimensional Gaussian,

$$h(i,j) = \frac{1}{(1 + i^2 + j^2)}, \tag{4.6}$$

truncated such that the function has a support of 9×9. The image restoration problem has the form (1.1), where $\rho = 0.0005$ and $A = HW$ are the composition of the 9×9 uniform blur matrix and the Haar discrete wavelet transform (DWT) operator. We compare the performance of MSGP with that of SGCS, SPARSA, and GPSR-BB by solving the problem and choose $\nu = 0$ in the MSGP method. As usual, we measure the quality of restoration by signal-to-noise ratio (SNR) defined as

$$\mathrm{SNR} = 10 \times \log_{10} \frac{\|x_{\mathrm{orig}}\|^2}{\|x_{\mathrm{orig}} - x\|^2}, \tag{4.7}$$

where x_{orig} and x are the original and restored images, respectively. We first run SGCS and stop the process if the following inequality is satisfied:

$$\frac{\|x^{k+1} - x^k\|}{\|x^k\|} < 10^{-5}, \tag{4.8}$$

and then run the other algorithms until their objective function value reach SGCS's value. Table 1 reports the number of iterations (Iter), the CPU time in seconds (Time), and the SNR to the original images (SNR).

Table 1: Test results for SGCS, SPARSA, GPSR-BB, SGP, and MSGP in image restoration.

Ima	SGCS			SPARSA			GPSR-BB			SGP			MSGP		
	Iter	Time	SNR	Iter	Time	SNR	Iter	Time	SNR	Iter	Time	SNR	Iter	Time	SNR
House	53	12.15	30.68	19	0.85	30.35	25	1.28	30.44	38	5.27	30.61	16	0.78	30.82
Cameraman	59	9.55	22.42	19	0.87	23.64	23	1.06	22.67	48	4.81	22.50	17	0.79	23.36
Barbara	150	234.05	22.95	29	7.08	23.72	41	12.78	22.93	62	42.12	23.06	26	6.92	23.09

It is easy to see from Table 1 that the MSGP is competitive with the well-known algorithms: SPARSA and GPSR-BB, in computing time and number of iterations and improves the SGCS greatly. Therefore we conclude that the MSGP provides a valid approach for solving ℓ_1-norm minimization problems arising from image restoration problems.

Preliminary numerical experiments show that SGP and MSGP algorithms have improved SGCS algorithm greatly. This may be because the system of nonsmooth equations solved here has lower dimension than that in [17] and the modification to projection steps that we made reduces the computational complexity.

Acknowledgments

The authors would like to thank the anonymous referee for the valuable suggestions and comments. L. Wu was supported by the NNSF of China under Grant 11071087; Z. Sun was supported by the NNSF of China under Grants 11126147 and 11201197.

References

[1] D. L. Donoho, “Compressed sensing,” *IEEE Transactions on Information Theory*, vol. 52, no. 4, pp. 1289–1306, 2006.

[2] M. Elad, B. Matalon, and M. Zibulevsky, “Image denoising with shrinkage and redundant representations,” in *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR ’06)*, pp. 1924–1931, New York, NY, USA, June 2006.

[3] S. Alliney and S. A. Ruzinsky, “An algorithm for the minimization of mixed ℓ_1 and ℓ_2 norms with application to Bayesian estimation,” *IEEE Transactions on Signal Processing*, vol. 42, no. 3, pp. 618–627, 1994.

[4] E. J. Candès, J. K. Romberg, and T. Tao, “Stable signal recovery from incomplete and inaccurate measurements,” *Communications on Pure and Applied Mathematics*, vol. 59, no. 8, pp. 1207–1223, 2006.

[5] D. L. Donoho, “For most large underdetermined systems of linear equations the minimal ℓ_1-norm solution is also the sparsest solution,” *Communications on Pure and Applied Mathematics*, vol. 59, no. 6, pp. 797–829, 2006.

[6] I. Daubechies, M. Defrise, and C. De Mol, “An iterative thresholding algorithm for linear inverse problems with a sparsity constraint,” *Communications on Pure and Applied Mathematics*, vol. 57, no. 11, pp. 1413–1457, 2004.

[7] C. De Mol and M. Defrise, “A note on wavelet-based inversion algorithms,” *Contemporary Mathematics*, vol. 313, pp. 85–96, 2002.

[8] A. Chambolle, R. A. DeVore, N. Y. Lee, and B. J. Lucier, “Nonlinear wavelet image processing: variational problems, compression, and noise removal through wavelet shrinkage,” *IEEE Transactions on Image Processing*, vol. 7, no. 3, pp. 319–335, 1998.

[9] A. Beck and M. Teboulle, “A fast iterative shrinkage-thresholding algorithm for linear inverse problems,” *SIAM Journal on Imaging Sciences*, vol. 2, no. 1, pp. 183–202, 2009.

[10] E. T. Hale, W. Yin, and Y. Zhang, “Fixed-point continuation for ℓ_1-minimization: methodology and convergence,” *SIAM Journal on Optimization*, vol. 19, no. 3, pp. 1107–1130, 2008.

[11] Z. Wen, W. Yin, D. Goldfarb, and Y. Zhang, “A fast algorithm for sparse reconstruction based on shrinkage, subspace optimization, and continuation,” *SIAM Journal on Scientific Computing*, vol. 32, no. 4, pp. 1832–1857, 2010.

[12] M. A. T. Figueiredo, R. D. Nowak, and S. J. Wright, “Gradient projection for sparse reconstruction: application to compressed sensing and other inverse problems,” *IEEE Journal of Selected Topics in Signal Processing*, vol. 1, no. 4, pp. 586–597, 2007.

[13] J. Barzilai and J. M. Borwein, “Two-point step size gradient methods,” *IMA Journal of Numerical Analysis*, vol. 8, no. 1, pp. 141–148, 1988.

[14] S. J. Wright, R. D. Nowak, and M. A. T. Figueiredo, “Sparse reconstruction by separable approximation,” *IEEE Transactions on Signal Processing*, vol. 57, no. 7, pp. 2479–2493, 2009.

[15] S. Yun and K.-C. Toh, “A coordinate gradient descent method for ℓ_1-regularized convex minimization,” *Computational Optimization and Applications*, vol. 48, no. 2, pp. 273–307, 2011.

[16] J. Yang and Y. Zhang, “Alternating direction algorithms for ℓ_1-problems in compressive sensing,” *SIAM Journal on Scientific Computing*, vol. 33, no. 1, pp. 250–278, 2011.

[17] Y. Xiao, Q. Wang, and Q. Hu, “Non-smooth equations based method for ℓ_1-norm problems with applications to compressed sensing,” *Nonlinear Analysis: Theory, Methods & Applications*, vol. 74, no. 11, pp. 3570–3577, 2011.

[18] L. Zhang and W. J. Zhou, “Spectral gradient projection method for solving nonlinear monotone equations,” *Journal of Computational and Applied Mathematics*, vol. 196, no. 2, pp. 478–484, 2006.

[19] Q. N. Li and D. H. Li, “A class of derivative-free methods for large-scale nonlinear monotone equations,” *IMA Journal of Numerical Analysis*, vol. 31, no. 4, pp. 1625–1635, 2011.

[20] M. V. Solodov and B. F. Svaiter, “A globally convergent inexact Newton method for systems of monotone equations,” in *Reformulation: Nonsmooth, Piecewise Smooth, Semismooth and Smoothing Methods*, M. Fukushima and L. Qi, Eds., vol. 22, pp. 355–369, Kluwer Academic Publishers, 1998.

[21] W. J. Zhou and D. H. Li, “Limited memory BFGS method for nonlinear monotone equations,” *Journal of Computational Mathematics*, vol. 25, no. 1, pp. 89–96, 2007.

[22] W. J. Zhou and D. H. Li, “A globally convergent BFGS method for nonlinear monotone equations without any merit functions,” *Mathematics of Computation*, vol. 77, no. 264, pp. 2231–2240, 2008.

[23] F. H. Clarke, *Optimization and Nonsmooth Analysis*, John Wiley & Sons, New York, NY, USA, 1983.

[24] L. Q. Qi and J. Sun, “A nonsmooth version of Newton’s method,” *Mathematical Programming A*, vol. 58, no. 3, pp. 353–367, 1993.

[25] X. Chen and S. Xiang, “Computation of error bounds for P-matrix linear complementarity problems,” *Mathematical Programming A*, vol. 106, no. 3, pp. 513–525, 2006.

[26] D. H. Li and M. Fukushima, “A modified BFGS method and its global convergence in nonconvex minimization,” *Journal of Computational and Applied Mathematics*, vol. 129, no. 1-2, pp. 15–35, 2001.