Nonmonotone Spectral Gradient Method for $\ell_1$-regularized Least Squares

Wanyou Cheng $^1$, Qingjie Hu $^{2,*}$

$^1$College of Computer, Dongguan University of Technology, China

$^2$School of Mathematics and Computing Science, Guilin University of Electronic Technology, China

(Received: 30 May 2016; Accepted: 10 August 2016)

Abstract  In the paper, we investigate a linear constraint optimization reformulation to a more general form of the $\ell_1$ regularization problem and give some good properties of it. We first show that the equivalence between the linear constraint optimization problem and the $\ell_1$ regularization problem. Second, the KKT point of the linear constraint problem always exists since the constraints are linear; we show that the half constraints must be active at any KKT point. In addition, we show that the KKT points of the linear constraint problem are the same as the stationary points of the $\ell_1$ regularization problem. Based on the linear constraint optimization problem, we propose a nonmonotone spectral gradient method and establish its global convergence. Numerical experiments with compressive sense problems show that our approach is competitive with several known methods for standard $\ell_2$-$\ell_1$ problem.

Keywords  $\ell_1$ minimization, Compressive sensing, Projection gradient

AMS 2010 subject classifications 90C06, 90C25,65Y20, 94A08

DOI: 10.19139/soic.v4i3.230

1. Introduction

In recent years, many studies focus on the following $\ell_1$ regularization problem:

$$
\min \phi(x) := f(x) + \mu \|x\|_1,
$$

where $f$ is continuously differentiable, $\mu$ is a given nonnegative regularization parameter and $\| \cdot \|_1$ is the one-norm. A particular case of (1) is the so-called $\ell_2$-$\ell_1$ problem

$$
\min_{x \in \mathbb{R}^n} \frac{1}{2} \|Ax - b\|_2^2 + \mu \|x\|_1,
$$

where $A \in \mathbb{R}^{m \times n}$ is dense (usually $m \leq n$), $b \in \mathbb{R}^m$ and $n$ is large, which has attracted much attention in signal/image denoising and data mining/classification [5, 9, 17].

Various types of algorithms have been proposed for solving (1). One of the most popular methods for solving problem (3) is the class of iterative shrinkage-thresholding algorithms (ISTA), where each iteration involves a matrix-vector multiplication involving $A$ and $A^T$ followed by a shrinkage/soft-threshold step, see, e.g., [13, 21]. To accelerate the convergence, a two-step ISTA (TWISTA) algorithm was developed in [6], the sequential subspace optimization techniques was added to ISTA [20], a faster shrinkage-thresholding algorithm, called
The first key step of our algorithm approach is to express (1) as a constraint optimization problem as in [26]. Specifically, the problem (1) can be transformed to the following problem with linear inequality constraints
\[
\min f(x) + \mu \sum_{i=1}^{n} u_i,
\]
\[
s.t. \left\{ \begin{array}{l}
  u_i + x_i \geq 0, \quad i = 1, \cdots, n, \\
  u_i - x_i \geq 0, \quad i = 1, 2, \cdots, n.
\end{array} \right.
\]

In this section, we shall show that the equivalence between the linear constraint optimization problem (3) and the \(\ell_1\) regularization problem (1). Second, the KKT point of the linear constraint problem always exists since the constraints are linear; we shall show that the constraint must be active at any KKT points. At last, we shall show that the KKT points of the linear constraint problem are the same as the stationary points of the \(\ell_1\) regularization problem.
The following theorem shows the equivalent between (1) and problem (3).

**Theorem 2.1**

If \( x^* \) is a solution of (1), then \( (|x^*|, x^*) \) is a solution of (3). Conversely, if \( (u^*, x^*) \) is a solution of (3), then \( x^* \) is a solution of (1).

**Proof**

Let \( \bar{x} \) be an any vector in \( R^n \) and choose \( \bar{u} \) such that \( (\bar{u}, \bar{x}) \) is a feasible point of (3). Then we have \( \bar{u}_i \geq |\bar{x}_i|, \) for \( i = 1, 2, \cdots, n. \) Since \( x^* \) is a solution of (1), we have

\[
    f(\bar{x}) + \mu \sum_{i=1}^{n} \bar{u}_i \geq f(\bar{x}) + \mu \sum_{i=1}^{n} |\bar{x}_i| \\
    \geq f(x^*) + \mu \sum_{i=1}^{n} |x^*_i|
\]

which shows that \( (|x^*|, x^*) \) is a solution of (3). On the other hand, suppose that \( (u^*, x^*) \) is a solution of (3) and \( x \in R^n \) is a any vector. We choose \( u \in R^n \) such that \( u_i \geq |x_i| \) for \( i = 1, 2, \cdots, n. \) Then \( (u, x) \) is a feasible point of (3) and we thus have

\[
    f(x) + \mu \sum_{i=1}^{n} u_i \geq f(x) + \mu \sum_{i=1}^{n} |x_i| \\
    \geq f(x^*) + \mu \sum_{i=1}^{n} u_i^* \\
    \geq f(x^*) + \mu \sum_{i=1}^{n} |x_i^*|.
\]

Then the first inequality and third inequality imply that \( x^* \) is a solution of (1).\[\Box\]

Since the constraints in (3) are all linear, the KKT point of the linear constraint problem (3) always exists. The following theorem about the first order necessary condition of (3) becomes obvious.

**Theorem 2.2**

Suppose that \( f : R^n \to R \) is continuously differentiable and \( z = (u, x) \) is a local solution of the constrained problem (3). Then there must exist multipliers \( \lambda^1 \in R^n \) and \( \lambda^2 \in R^n \) such that the following KKT conditions hold:

\[
\begin{align*}
    \nabla f(x) - \lambda^1_i + \lambda^2_i &= 0, \quad i = 1, 2, \cdots, n, \\
    \mu - \lambda^1_i - \lambda^2_i &= 0, \quad i = 1, 2, \cdots, n, \\
    \min \left\{ \lambda^1_i, u_i + x_i \right\} &= 0, \quad i = 1, 2, \cdots, n, \\
    \min \left\{ \lambda^2_i, u_i - x_i \right\} &= 0, \quad i = 1, 2, \cdots, n.
\end{align*}
\]

(4)

The following theorem show that at any KKT point of (3), either constraint \( u_i - x_i \geq 0 \) or \( u_i + x_i \geq 0 \) must be active for any \( i = 1, 2, \cdots, n. \) That is, the half constraints of (3) must be active at any KKT point.

**Theorem 2.3**

Suppose that \( f : R^n \to R \) is continuously differentiable and \( (u, x, \lambda^1, \lambda^2) \) is a KKT point of the constrained problem (3). Then the equality \( u_i = |x_i| \) holds for all \( i = 1, 2, \cdots, n. \) Moreover, we have

\[ x_i = 0 \iff u_i = 0. \]

**Proof**

We first prove that \( u_i = |x_i| \) holds for all \( i \) with \( x_i = 0. \) By the constraint condition \( -u_i \leq x_i \leq u_i, \) we get \( x_i = 0 \)
if \( u_i = 0 \). Suppose that \( x_i = 0 \) for some \( i \). By the KKT condition (4), we get that for all \( i = 1, 2, \cdots, n \),
\[
\lambda_i^1 u_i = 0 \quad \text{and} \quad \lambda_i^2 u_i = 0.
\]
Furthermore, by \( \mu = \lambda_i^1 + \lambda_i^2 \), we get
\[
\mu u_i = \lambda_i^1 u_i + \lambda_i^2 u_i = 0,
\]
which shows \( u_i = 0 \). If there exists an index \( i \) with \( x_i \neq 0 \) such that \( u_i > |x_i| > 0 \). By the KKT condition, we have \( \lambda_i^1 = 0 \) and \( \lambda_i^2 = 0 \), which contradicts the condition \( \mu = \lambda_i^1 + \lambda_i^2 > 0 \).

A point \( x \) is called a stationary point of (1) if it satisfies
\[
\begin{cases}
\nabla f_i(x) + \mu = 0 & \text{if } x_i > 0, \\
\nabla f_i(x) - \mu = 0 & \text{if } x_i < 0, \\
|\nabla f_i(x)| \leq \mu & \text{if } x_i = 0,
\end{cases}
\]
for \( i = 1, 2, \cdots, n \). The following theorem show that the KKT points of the linear constraint problem (3) are the same as the stationary points of the \( \ell_1 \) regularization problem.

**Theorem 2.4**
Suppose that \( f : \mathbb{R}^n \to \mathbb{R} \) is continuously differentiable and \((u, x, \lambda^1, \lambda^2)\) is a KKT point of the constrained problem (3). Then \( x \) is a stationary point of (1). Conversely, if \( x \) is a stationary point of (1), then there exist multipliers \( \lambda^1 \) and \( \lambda^2 \) such that \((|x|, x, \lambda^1, \lambda^2)\) is a KKT point of (3).

**Proof**
Suppose that \((u, x, \lambda^1, \lambda^2)\) is a KKT point of (3). By Theorem 2.3, we have \( u_i = |x_i| \) for all \( i = 1, 2, \cdots, n \). By the first, third and fourth equality of (4), we have
\[
\lambda_i^1 \geq 0, \quad \lambda_i^2 \geq 0 \quad \text{and} \quad |\nabla f_i(x)| = |\lambda_i^1 - \lambda_i^2| \leq \mu.
\]
If \( x_i > 0 \), then \( u_i = x_i \). By the second, third, fourth inequality of (4), we get \( \lambda_i^1 = 0 \) and \( \lambda_i^2 = \mu \). By the first inequality of (4), we get \( \nabla f_i(x) + \mu = 0 \). If \( x_i < 0 \), then \( u_i = -x_i \). By the second, third and fourth inequality of (4), we get \( \lambda_i^2 = 0, \lambda_i^1 = \mu \) and \( \nabla f_i(x) - \mu = 0 \). Thus, \( x \) is a stationary point of (1). Conversely, suppose that \( x \) is a stationary point of (1). We let \( u_i = |x_i| \) for all \( i = 1, 2, \cdots, n \). If \( x_i > 0 \), then we have \( u_i = x_i \) and \( \nabla f_i(x) + \mu = 0 \). In this case, we let \( \lambda_i^1 = 0 \) and \( \lambda_i^2 = \mu \) which satisfy (5). If \( x_i < 0 \), then we have \( u_i = -x_i \) and \( \nabla f_i(x) - \mu = 0 \). In this case, we let \( \lambda_i^1 = \mu \) and \( \lambda_i^2 = 0 \) which satisfy (5). If \( x_i = 0 \), we have \( u_i = 0 \), it suffices to let \( \lambda_i^1 \) and \( \lambda_i^2 \) be any positive constants that satisfy \( \lambda_i^1 + \lambda_i^2 = \mu \). Consequently, the KKT conditions (4) hold.

3. **Algorithm and Convergence Result**

3.1. **Algorithm**
In this section, we begin with some notation. Let \( \Omega = \{z = (u, x)^T | u_i + x_i \geq 0 \text{ and } u_i - x_i \geq 0, \ i = 1, 2, \cdots, n, u \in \mathbb{R}^n, x \in \mathbb{R}^n \} \) and \( P_{\Omega}(v) \) denote the projection of any vector \( v \) on the set \( \Omega \). We first gives the explicit form of \( P_{\Omega}(v) \). Then, we propose the algorithm and give some convergence result. The following theorem shows that \( \Omega \) is a nonempty closed convex set.

**Theorem 3.1**
The set \( \Omega \) is a nonempty closed convex set.

**Proof**
Clearly, the set \( \Omega \) is a nonempty closed set. For any \( z_1 = (u_1, y_1)^T \in \mathbb{R}^{2n} \), \( z_2 = (u_2, y_2)^T \in \mathbb{R}^{2n} \) and \( \alpha \in (0, 1) \).
Then we have
\[
\begin{align*}
&\begin{cases}
  u_1 + y_1 \geq 0, \\
  u_1 - y_1 \geq 0, \\
  u_2 + y_2 \geq 0, \\
  u_2 - y_2 \geq 0.
\end{cases}
\end{align*}
\]
Then, we get \(\alpha(u_1 + y_1) + (1 - \alpha)(u_2 + y_2) \geq 0\) and \(\alpha(u_1 - y_1) + (1 - \alpha)(u_2 - y_2) \geq 0\), which shows that
\(\alpha z_1 + (1 - \alpha)z_2 \in \Omega\).

Above theorem shows that \(\Omega\) is a nonempty closed convex set. Thus, we can compute the projection of any vector \(v\) on the set \(\Omega\). The following theorem gives the explicit form of \(P_\Omega(v)\).

**Theorem 3.2**
Consider the optimal problem
\[
P_\Omega(z) = \arg \min_{v \in \Omega} \frac{1}{2} \|z - v\|^2
\]
where \(z = (z_1, z_2)^T\), \(P_\Omega(z) = (u, x)^T\) and \(z_1, z_2, u, x \in \mathbb{R}^n\). Then, for \(i = 1, 2, \ldots, n\), we get that

1. if \(\{z_1(i) + z_2(i) \geq 0, z_1(i) - z_2(i) \geq 0\}\), then
   \[
   \begin{cases}
   u(i) = z_1(i) \\
   x(i) = z_2(i).
   \end{cases}
   \]
2. if \(\{z_1(i) + z_2(i) < 0, z_1(i) - z_2(i) < 0\}\), then
   \[
   \begin{cases}
   u(i) = 0 \\
   x(i) = 0.
   \end{cases}
   \]
3. if \(\{z_1(i) + z_2(i) \geq 0, z_1(i) - z_2(i) < 0\}\), then
   \[
   \begin{cases}
   u(i) = \frac{z_1(i) + z_2(i)}{2} \\
   x(i) = \frac{z_1(i) + z_2(i)}{2}.
   \end{cases}
   \]
4. if \(\{z_1(i) + z_2(i) < 0, z_1(i) - z_2(i) \geq 0\}\), then
   \[
   \begin{cases}
   u(i) = \frac{z_1(i) - z_2(i)}{2} \\
   x(i) = \frac{z_2(i) - z_1(i)}{2}.
   \end{cases}
   \]

**Proof**
Define the Lagrange function of (6)
\[
L(u, x, \rho^1, \rho^2) = \frac{1}{2} \|z_1 - u\|^2 + \frac{1}{2} \|z_2 - x\|^2 - \sum_{i=1}^{n} \rho^1_i (u_i + x_i) - \sum_{i=1}^{n} \rho^2_i (u_i - x_i)
\]
where $\rho^1 \in \mathbb{R}^n$ and $\rho^2 \in \mathbb{R}^n$ are multipliers. The KKT condition for (6) is as follows

$$
\begin{aligned}
\nabla L_x(u, x, \rho^1, \rho^2)_i &= x_i - (z_2)_i - \rho^1_i + \rho^2_i = 0, \ i = 1, 2, \cdots, n, \\
\nabla L_u(u, x, \rho^1, \rho^2)_i &= u_i - (z_1)_i - \rho^1_i - \rho^2_i = 0, \ i = 1, 2, \cdots, n, \\
\min(\rho^1_i, u_i + x_i) &= 0, \ i = 1, 2, \cdots, n, \\
\min(\rho^2_i, u_i - x_i) &= 0, \ i = 1, 2, \cdots, n.
\end{aligned}
$$

(7)

Note that (6) is a convex optimization problem. Thus for any solution $(u, x, \rho^1, \rho^2)$ of (7), $(u, x)$ is a solution of (6). We consider the following four cases. Case (i) if

$$
\begin{aligned}
\{ & z_1(i) + z_2(i) \geq 0 \\
& z_1(i) - z_2(i) \geq 0,
\end{aligned}
$$

then we let $\rho^1_i = \rho^2_i = 0$, $u(i) = z_1(i)$ and $x(i) = z_2(i)$ which satisfy (7).

Case (ii) if

$$
\begin{aligned}
\{ & z_1(i) + z_2(i) < 0 \\
& z_1(i) - z_2(i) < 0,
\end{aligned}
$$

then we let $\rho^1_i = \frac{z_1(i) + z_2(i)}{2}$, $\rho^2_i = \frac{z_2(i) - z_1(i)}{2}$ and $u(i) = x(i) = 0$ which satisfy (7).

Case (iii) if

$$
\begin{aligned}
\{ & z_1(i) + z_2(i) \geq 0 \\
& z_1(i) - z_2(i) < 0,
\end{aligned}
$$

then we let $\rho^1_i = 0$, $\rho^2_i = \frac{z_2(i) - z_1(i)}{2}$ and $u(i) = x(i) = \frac{z_1(i) + z_2(i)}{2}$ which satisfy (7).

Case (iv) if

$$
\begin{aligned}
\{ & z_1(i) + z_2(i) < 0 \\
& z_1(i) - z_2(i) \geq 0,
\end{aligned}
$$

then we let $\rho^1_i = -\frac{z_1(i) + z_2(i)}{2}$, $\rho^2_i = 0$, $u(i) = \frac{z_1(i) - z_2(i)}{2}$ and $x(i) = -\frac{z_1(i) + z_2(i)}{2}$ which satisfy (7).

Based on the above discussion, we propose the projection gradient method for (1) as follows.

**Algorithm 1. (Projection Gradient Method)**

Step 0. Given an initial point $z^0 = (u^0, x^0) \in \Omega$ and positive constants $M$, $\alpha_{\min}$, $\alpha_{\max}$, $\eta$ and $\delta \in (0, 1)$. Set $k := 0$.

Step 1. Perform the convergence test and terminate with an approximate solution $z^k$ if the stopping criterion is satisfied.

Step 2. Choose $\theta^k \in [\alpha_{\min}, \alpha_{\max}]$ and compute $d^k = P_{\Omega}(z^k - \theta^k \nabla \phi(z^k)) - z^k$.

Step 3. Determine $\alpha^k := \max\{\eta^j, j = 0, 1, \cdots\}$ satisfying

$$
\phi(z^k + \alpha^k d^k) \leq \phi^k_{\max} + \delta(\nabla \phi(z^k))^T d^k,
$$

where $\phi^k_{\max} = \max\{\phi(z^{k-j}) : 0 \leq j < \min(k, M - 1)\}$.

Step 4. Let the next iterate be $z^{k+1} := z^k + \alpha^k d^k$.

Step 5. Set $k := k + 1$ and go to Step 1.

To accelerate the projection gradient method, we shall apply the Barzilar-Borwein steplength to Algorithm 1. To this aim, we briefly recall the Barzilar-Borwein method (for example, see [3, 15]). Consider the unconstrained minimization problem

$$
\min_{x \in \mathbb{R}^n} G(x).
$$

where $G : \mathbb{R}^n \to \mathbb{R}$ is continuously differentiable. The Barzilai-Borwein method is defined by

$$
x^{k+1} = x^k - \alpha^{k}_{BB} \nabla G(x^k),
$$

where the scalar $\alpha^k_{BB}$ is given by
\begin{equation}
\alpha^k_{BB} = \frac{\|s^{k-1}\|^2}{(s^{k-1})^T y^{k-1}}, \tag{8}
\end{equation}
where $\alpha^k_{BB}$ is called the Barzilai-Borwein steplength [3], $s^{k-1} = x^k - x^{k-1}$ and $y^{k-1} = \nabla G(x^k) - \nabla G(x^{k-1})$.

Due to its easy implementation, efficiency and low storage requirement, BB-type methods have widely been used in many applications such as box constrained optimization [7, 14], nonlinear equations [12] and sparse reconstruction [35, 22]. The Barzilai-Borwein steplength used in Algorithm 1 is defined by
\begin{equation}
\theta^k_{BB} = \frac{\|z^{k+1} - z^k\|^2}{(\nabla \phi(z^{k+1}) - \nabla \phi(z^k))^T (z^{k+1} - z^k)}.
\end{equation}

Note that
\[\nabla \phi_s(z^k) = \nabla f(x^k) \text{ and } \nabla \phi_u(z^k) = \mu(1, 1, \ldots, 1)^T\]
is a constant. Thus, we get
\begin{equation}
\theta^k_{BB} = \frac{\|z^{k+1} - z^k\|^2}{(\nabla f(x^{k+1}) - \nabla f(x^k))^T (x^{k+1} - x^k)} = \frac{\|u^{k+1} - u^k\|^2 + \|x^{k+1} - x^k\|^2}{(\nabla f(x^{k+1}) - \nabla f(x^k))^T (x^{k+1} - x^k)}.
\end{equation}

To avoid small and large values of $\theta^k_{BB}$, we project it in the interval $[\alpha_{min}, \alpha_{max}]$, where $\alpha_{min} < \alpha_{max}$ are given positive constants. That is, we let
\begin{equation}
\theta^k_{BB} = \min\{\alpha_{max}, \max\{\alpha_{min}, \theta^k_{BB}\}\}. \tag{9}
\end{equation}

For simplicity, we call Algorithm (1) with the steplength (9) used in step 2 of Algorithm (1) as the projection Barzilai-Borwein algorithm and abbreviate it as PBB.

### 3.2. Convergence Result

In this section, we analyze the convergence of Algorithm 1. To this aim, we make the following assumptions on the objective function.

**Assumption 3.1**

(i) The level set $L(z^0) = \{z = (u, x)^T \in \mathbb{R}^{2n} : \phi(z) \leq \phi(z^0)\}$ is bounded.

(ii) $f$ has continuous partial derivative on an open set that contains the level set $L(z^0)$.

The following theorem shows that every accumulation point of $\{z^k\}$ is a stationary point of (3). The proof of the following theorem is similar to the one in [7] and hence was omitted.

**Theorem 3.3**

Assume that $\phi$ satisfies Assumption 3.1. Let $\{z^k\}$ be the sequence generated by Algorithm 1. If $d^k \neq 0$ for all $k$, every accumulation point $z^*$ of $\{z^k\}$ is a stationary point of (3). Moreover, if $f$ is convex, then every accumulation point $z^*$ of $\{z^k\}$ is a solution of (3).

### 4. Numerical Experiments

In this section, we do some numerical experiments to test the performance of the proposed method and compare it with the following three existing solvers, $\ell_1$-LS [26] FPC_AS [33] and GPSR_BB [22]. All codes are written in MATLAB 7.0 and all tests described in this section were performed on a PC with Intel I5-3230 2.6GHZ CPU processor and 4G RAM memory with a Windows operating system.

Experiments in [22, 24, 33, 35] have confirmed the effectiveness of continuation. Therefore, we embedded our method in an adaptive continuation procedure. Specifically, we use the adaptive continuation procedure in [35]
which was also used in GPSR\_BB [22]. We implemented Algorithm 1 with the following parameters \( M = 5, \alpha_{\text{min}} = 10^{-10}, \alpha_{\text{max}} = 10^{10}, \delta = 10^{-2} \) and \( \eta = 0.5 \) and implement the continuation procedure with the parameter \( \varsigma = 0.2 \). The initial point of all tested algorithms is the zero vector. The other three algorithms were run with default parameters.

In Table 1, we summarize a list of symbols used in the subsequent tables and figures.

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>m, n</td>
<td>numbers of rows and columns of ( A ), respectively</td>
</tr>
<tr>
<td>cpu</td>
<td>cpu time</td>
</tr>
<tr>
<td>nnzx</td>
<td>number of the nonzeros in the recovered solution</td>
</tr>
<tr>
<td>nMat</td>
<td>total number of matrix-vector products involving ( A ) and ( A^T )</td>
</tr>
<tr>
<td>MSE</td>
<td>the relative error between the recovered solution ( x ) and the exact sparsest solution ( x_s ), i.e., ( \text{MSE} = \frac{|x - x_s|}{|x_s|} )</td>
</tr>
</tbody>
</table>

4.1. \( \ell_2\ell_1 \) problem

In this subsection, we consider a typical compressed sensing scenario, that is the problem (2), where the goal is to reconstruct a length-\( n \) sparse signal from \( m \) observations, where \( m < n \). The \( m \times n \) measure matrix \( A \) is obtained by first filling it with independent samples of a standard Gaussian distribution and then orthonormalizing the rows. These random matrices are generated by using MATLAB command \texttt{randn}. To generate the signal \( x_s \), we first generated the support by randomly selecting \( T \) indices between 1 and \( n \) and then assigned a value to \( x_i \) for each \( i \) in the support by one of the following four methods:

- Type 1: one (zero-one signal);
- Type 2: the sign of a normally distributed random variable;
- Type 3: a normally distributed random variable (Gaussian signal);
- Type 4: a uniformly distributed random variable \((-1, 1)\);

In this experiment, we tested the matrix \( A \) with size \( n = 4096 \) and \( m = \text{round}(0.1 \times n) \) or \( m = \text{round}(0.2 \times n) \) and considered a range of degrees of sparseness: the number \( T \) of nonzero spikes in \( x_s \) ranges from 1 to 30 for each type of the elements in the support. The observation \( b \) is generated by \( b = Ax_s \) and the regularization parameter \( \mu \) is taken as \( \mu = 0.05\|A^Tb\|_\infty \). The above procedure yields a total of 240 problems. For each data set \((x_s, A, b)\), we first ran \( \ell_1 \ell_s \) and stored the final value of the objective function and then ran the other algorithms until they reach the same objective function value.

Each component of signal \( x_s \) and the final solution obtained by each tested method is considered as a nonzero component when its absolute value is greater than \( 0.001\|x_s\|_\infty \). We adopt the performance profiles by Dolan and Moré [19] to evaluate the CPU time and the numbers of MSE, nMat and nnz. Figures 1-8 show the performance profiles of the five methods relative to CPU time and the numbers of MSE, nMat and nnz. It shows that the PBB method performs best for the 240 test problems and generally requires less CPU time and fewer numbers of nMat and obtain less MSE and the same nnz as other algorithms.
4.2. Group-separable regularizer

In this subsection, we examine the performance of the proposed methods using the group separable regularizers [35] where

$$\phi(x) = \frac{1}{2} \|Ax - b\|^2 \quad \text{and} \quad \psi(x) = \mu \sum_{i=1}^{n} \|x[i]\|_1,$$

where $x[1], x[2], \cdots, x[m]$, are $m$ disjoint subvectors of $x$ and $A \in \mathbb{R}^{1024 \times 4096}$ was obtained by the same way as that in subsection 4.1. The vector $x_s$ has 4096 components, divided into $k = 64$ groups of length $l_i = 64$. To generate $x_s$, we randomly chose from one to eight groups and filled them with zero-mean Gaussian random samples of unit variance, while all the other groups are filled with zeros. The target vector is $b = Ax_s + e$, where the noise $e$ is a Gaussian noise with mean zero and variance $10^{-4}$. The regularization parameter is chosen as suggested in [35]: $\mu = 0.05 \|A^T b\|_\infty$. We used the same stopping criterion as that in Section 4.1. We ran 10 test problems and gives the average CPU time needed by the four methods in Table 2.
NONMONOTONE SPECTRAL GRADIENT METHOD FOR \(\ell_1\)-REGULARIZED LEAST SQUARES

Figure 5. Performance profiles based on CPU time in log2 scale for \(m = \text{round}(0.1n)\)

Figure 6. Performance profiles based on MSE in log2 scale for \(m = \text{round}(0.1n)\)

Figure 7. Performance profiles based on nmat for \(m = \text{round}(0.1n)\)

Figure 8. Performance profiles based on nnz for \(m = \text{round}(0.1n)\)

Table 2. Statical data

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>(\ell_1) ls</th>
<th>FPC_AS</th>
<th>GPSR_BB</th>
<th>PBB</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU time</td>
<td>2.3494</td>
<td>0.4193</td>
<td>0.2925</td>
<td>0.2399</td>
</tr>
</tbody>
</table>

From Table 2, we can observe that PBB is much faster than the \(l_1\)ls method and are comparable to the FPC\_AS and GPSR\_BB methods.

4.3. Image deblurring problem

In this subsection, we present results for one image restoration problems referred to as Cameraman (see Figure 7). The images are \(256 \times 256\) grayscale images; that is, \(n = m = 256^2 = 65536\). The image restoration problem has the form (3), where \(\mu = 0.00005\). We used the same stopping criterion as that in Section 4.1. In the test problem, the FPC\_AS method fails to satisfy the stopping condition. So we do not report the CPU time and the obtained image of it. Table 3 reports the average CPU time. The results in Table 3 and Figure 9 again indicate that PBB yields much better performance for the test problem.

Table 3. Statical data

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>(l_1) ls</th>
<th>GPSR_BB</th>
<th>PBB</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU time</td>
<td>137.34</td>
<td>1.81</td>
<td>1.51</td>
</tr>
</tbody>
</table>
5. Conclusion

In the paper, we investigate a linear constraint optimization reformulation to a more general form of the $\ell_1$ regularization problem and given some good properties of it. We first show that the equivalence between the linear constraint optimization problem and the $\ell_1$ regularization problem. Second, the KKT point of the linear constraint problem always exists since the constraints are linear; we show that the half constraints must be active at any KKT point. In addition, we show that the KKT points of the linear constraint problem are the same as the stationary points of the $\ell_1$ regularization problem. Based on the linear constraint optimization problem, we propose a nonmonotone spectral gradient Method. Under appropriate conditions, we showed that the method is globally convergent. The numerical results in Section 4 demonstrated the effectiveness of the algorithm for solving some standard $\ell_2-\ell_1$ problems.

Acknowledgment

This work was partly supported by the Chinese NSF Grant (no. 11371154 and 11461015), Guangxi Natural Science Foundation (no. 2015GXNSFAA139010) and by the Guangdong Province Outstanding Young Teacher Training Program (no. 3XZ150603 and 2015KTSCX1).
REFERENCES


