J Sci Comput (2014) 61:17–41
DOI 10.1007/s10915-013-9815-8

Nonmonotone Barzilai–Borwein Gradient Algorithm for ℓ_1-Regularized Nonsmooth Minimization in Compressive Sensing

Yunhai Xiao · Soon-Yi Wu · Liqun Qi

Received: 1 November 2012 / Revised: 3 September 2013 / Accepted: 28 December 2013 /
Published online: 11 January 2014

Abstract This study aims to minimize the sum of a smooth function and a nonsmooth ℓ_1-regularized term. This problem as a special case includes the ℓ_1-regularized convex minimization problem in signal processing, compressive sensing, machine learning, data mining, and so on. However, the non-differentiability of the ℓ_1-norm causes more challenges especially in large problems encountered in many practical applications. This study proposes, analyzes, and tests a Barzilai–Borwein gradient algorithm. At each iteration, the generated search direction demonstrates descent property and can be easily derived by minimizing a local approximal quadratic model and simultaneously taking the favorable structure of the ℓ_1-norm. A nonmonotone line search technique is incorporated to find a suitable stepsize along this direction. The algorithm is easily performed, where each iteration requiring the values of the objective function and the gradient of the smooth term. Under some conditions, the proposed algorithm appears globally convergent. The limited experiments using some nonconvex unconstrained problems from the CUTEr library with additive ℓ_1-regularization illustrate that the proposed algorithm performs quite satisfactorily. Extensive experiments for ℓ_1-regularized least squares problems in compressive sensing verify that our algorithm compares favorably with several state-of-the-art algorithms that have been specifically designed in recent years.

Y. Xiao (✉)
Institute of Applied Mathematics, College of Mathematics and Information Science,
Henan University, Kaifeng 475000, China
e-mail: yhxiao@henu.edu.cn

S.-Y. Wu
National Center for Theoretical Sciences (South), National Cheng Kung University,
Tainan 700, Taiwan
e-mail: soonyi@mail.ncku.edu.tw

L. Qi
Department of Applied Mathematics, Hong Kong Polytechnic University,
Hung Hom, Kowloon, Hong Kong
e-mail: maqilq@polyu.edu.hk

Keywords Nonsmooth optimization · Nonconvex optimization · Barzilai–Borwein gradient algorithm · Nonmonotone line search · ℓ_1 regularization · Compressive sensing

Mathematics Subject Classification 65L09 · 65K05 · 90C30 · 90C25

1 Introduction

The focus of this paper is on the following structured minimization:

$$\min_{x\in\mathbb{R}^n} F(x) := f(x) + \mu\|x\|_1, \tag{1.1}$$

where $f : \mathbb{R}^n \to \mathbb{R}$ is a continuously differentiable (may be nonconvex) function that is bounded below, $\|\cdot\|_1$ is the ℓ_1-norm of a vector, and parameter $\mu > 0$ is used to trade off both terms for minimization. Given its structure, problem (1.1) covers a wide range of apparently related formulations in different scientific fields, including linear and logistic regression [45], compressive sensing [18], sparse inverse covariance estimation [46], and sparse principal component analysis [27,42].

1.1 Problem Formulations

A special case of model (1.1) is the ℓ_1-norm regularized least squares problem shown below

$$\min_{x\in\mathbb{R}^n} \frac{1}{2}\|Ax - b\|_2^2 + \mu\|x\|_1, \tag{1.2}$$

where $A \in \mathbb{R}^{m\times n}$ $(m \ll n)$ is a linear operator, and $b \in \mathbb{R}^m$ is an observation. Model (1.2) mainly appears in compressive sensing, which is an emerging methodology in digital signal processing, and has attracted intensive research activities over the past years [8–11,18]. Compressive sensing is based on the fact that if the original signal is sparse or approximately sparse in some orthogonal basis, an exact restoration can be produced by solving problem (1.2).

Another prevalent case of (1.1) that has attracted much interest in machine learning is the linear and logistic regression. Assume the training date $A = [a_1, \ldots, a_m]^\top \in \mathbb{R}^{m\times n}$ and class labels $y \in \{-1, +1\}^m$. A linear classifier is a hyperplane $\{w_i : x^\top a_i + b = 0\}$, where $x \in \mathbb{R}^n$ is a set of weights, and $b \in \mathbb{R}$ is the intercept. A frequently used model is the ℓ_2-loss support vector machine, which is given by

$$\min_{x\in\mathbb{R}^m, b\in\mathbb{R}} \sum_{i=1}^{m} \max\left\{0, 1 - y_i(x^\top a_i + b)\right\}^2 + \mu\|x\|_1. \tag{1.3}$$

The ℓ_2-lose function is continuous, but not differentiable because of the "max" operation. In the logistic model, the probability distribution of the classifier label y, given a_i, has the form

$$p(y_i|a_i) = \frac{1}{1 - e^{-(x^\top a_i + b)y_i}}.$$

The weights x and the intercept b can be found by minimizing the average loss as

$$\min_{x\in\mathbb{R}^n, b\in\mathbb{R}} \frac{1}{m}\sum_{i=1}^{m} \phi\left(\left(x^\top a_i + b\right) y_i\right),$$

where $\phi(z) = \log(1 + e^{-z})$ is the logistic loss function. Finding the maximum likelihood estimate of x and b is called logistic regression. To derive a sparse vector x, the ℓ_1-regularized logistic regression can be formulated as

$$\min_{x\in\mathbb{R}^n, b\in\mathbb{R}} \frac{1}{m}\sum_{i=1}^{m} \log\left(1 + e^{-(x^\top a_i + b)y_i}\right) + \mu\|x\|_1. \tag{1.4}$$

Obviously, the logistic loss function is twice differentiable.

Although the models of these problems have similar structures, they may be very different from a real-data point of view. For example, in compressive sensing, the length of measurement m is much smaller than that of the original signal ($m \ll n$), and the encoding matrix A is dense. However, in machine learning, the numbers of instance m and features n are both large, and the data A is very sparse.

1.2 Existing Algorithms

As the ℓ_1-regularized term is non-differentiable when x contains zero values, the use of the standard unconstrained smooth optimization tools are generally precluded. In the past decades, various approaches have been proposed, analyzed, and implemented in compressive sensing and machine learning literature. One approach involves a variety of algorithms for special cases where in $f(x)$ has a specific functional form, such as the least squares (1.2), the square loss (1.3), and the logistic loss (1.4). In the following, we briefly review some of these approaches in the literature.

The first popular approach falls into the coordinate descent method. At the current iterate x_k, the simple coordinate descent method updates one component at a time to generate x_k^j, $j = 1, \ldots, n+1$ such that $x_k^1 = x_k$, $x_k^{n+1} = x_{k+1}$ and solves a one-dimensional subproblem

$$\min_z \quad F\left(x_k^j + ze^j\right) - F\left(x_k^j\right), \tag{1.5}$$

where e^j is defined as the jth column of an identity matrix. Clearly, the objective function has one variable and one non-differentiable point at $z = -e^j$. To solve the logistic regression model (1.4), Bayesian binary regression (BBR) [21] solves the sub-problem approximately using the trust region method with Newton step; coordinate descent with Newton step (CDN)[12] improves the performance of BBR by applying a one-dimensional Newton method and a line search technique. Instead of cyclically updating one component at a time, the stochastic coordinate descent method [35] randomly selects the working components to achieve enhanced performance; the block Coordinate Gradient Descent (CGD) algorithm [37,47] is based on the approximated convex quadratic model for f and selects the working variables according to some rules.

The second type of approach is to transform model (1.1) into an equivalent box-constrained optimization problem by variable splitting. Let $x = u - v$ with $u_i = \max\{0, x_i\}$ and $v_i = \max\{0, -x_i\}$. Then, model (1.1) can be reformulated equivalently as

$$\min_{u,v} f(u - v) + \mu\sum_{i=1}^{n}(u_i + v_i), \quad \text{s.t. } u \geq 0, \quad v \geq 0. \tag{1.6}$$

The objective function and constraints are smooth. Therefore, the model can be solved by any standard box-constrained optimization technique. However, an obvious drawback of this

approach is that it doubles the number of variables. Gradient projection for sparse reconstruction (GPSR) [20] solves (1.6) and subsequently solves (1.2) using the Barzilai–Borwein gradient method [2] with an efficient nonmonotone line search [22]. The method is actually an application of the well-known spectral projection gradient [5] in compressive sensing. The trust region Newton algorithm [26,45] minimizes (1.6) then solves the logistic regression model (1.4) and exhibits powerful vitality through a series of comparisons. To solve (1.3) and (1.4), the interior-point algorithm [24,25] forms a sequence of unconstrained approximations by appending a "barrier" function to the objective function (1.6), thereby ensuring that u and v remain sufficiently positive. Moreover, truncated Newton steps and preconditioned conjugate gradient iterations are used to produce the search direction.

The third type of method approximates the ℓ_1-regularized term with a differentiable function. The simple approach replaces the ℓ_1-norm with a sum of multi-quadric functions

$$l(x) \triangleq \sum_{i}^{n} \sqrt{x_i^2 + \epsilon},$$

where ϵ is a small positive scalar. This function is twice-differentiable, and $\lim_{\epsilon \to 0^+} l(x) = \|x\|_1$. Subsequently, several smooth unconstrained optimization approaches can be applied based on this approximation. However, the performance of these algorithms is highly influenced by the parameter values, and the condition number of the corresponding Hessian matrix becomes large as ϵ decreases. Nesterov's smoothing technique [28] is used to construct smooth functions to approximate some specific structured convex nonsmooth function. Based on this technique, NESTA (Nesterov's Algorithm) [4] solves problem (1.2) using first-order gradient information.

The fourth type of approach falls into the subgradient-based Newton-type algorithm. An important attempt in this class is that of Andrew and Gao [1], who extend the well-known limited memory BFGS method [30] to solve ℓ_1-regularized logistic regression model (1.4) and propose an orthant-wise limited memory quasi-Newton method. At each iteration, this method computes a search direction over an orthant containing the previous point. The subspace BFGS method [44] involves an inner iteration approach to find the descent quasi-Newton direction and subgradient Wolfe conditions to determine the stepsize that ensures decreasing the objective functions. This method is globally convergent and capable of solving general nonsmooth convex minimization problems.

Aside from GPSR and NESTA, other specially designed solvers are available. Using an operator splitting technique, Hale, Yin, and Zhang derive the iterative shrinkage/thresholding (IST) fixed-point continuation algorithm (FPC) [23]. By combining the interior-point algorithm [24], FPC is also extended to solve large-scale ℓ_1-regularized logistic regression [36]. A closely related algorithm to FPC is the fixed-point continuation and active set FPC_AS [39,40], which solves a smooth subproblem to determine the magnitudes of the nonzero components of x based on an active set. Two-step IST algorithm (TwIST) [6] and fast IST algorithm (FISTA) [3] speed up the performance of IST algorithm and have virtually the same complexity but with better convergence properties. Another closely related method is the sparse reconstruction algorithm (SpaRSA) [41], which involves minimizing a non-smooth convex problem with separable structures. The spectral projected gradient algorithm named SPGL1 [38] solves the lasso model (1.2) by the spectral gradient projection method with an efficient Euclidean projection on ℓ_1-norm ball. The alternating directions method called YALL1 [43] investigates ℓ_1-norm problems from either the primal or the dual forms and solves ℓ_1-regularized problems of different types.

All the reviewed algorithms differ in various aspects such as convergence speed, ease of implementation, and practical applicability. No evidence can verify that which algorithm outperforms the others under all scenarios.

1.3 Contributions and Organization

Although much progress has been achieved in solving problem (1.1), existing algorithms mainly deal with cases where f is a convex function and even least squares. In this study we propose a Barzilai–Borwein gradient algorithm to solve ℓ_1-regularized nonsmooth minimization problems. At each iteration, we approximate f locally by a convex quadratic model, where the Hessian is replaced by the multiples of a spectral coefficient with an identity matrix. The search direction is determined by minimizing the quadratic model and maximizing the use of the ℓ_1-norm structure. We show that the generated direction contains the one in FPC_AS [40] as a special case and is descent, which guarantees the existence of a positive stepsize along the direction. In our algorithm, we adopt the nonmonotone line search of Grippo, Lampariello, and Lucidi [22], which allows the function values to increase occasionally in some iteration but decrease in the whole iterative process. The nonmonotone line search is attractive because it saves a considerable number of function evaluations, which are the computational burden in large datasets. The method is easily performed, as only the value of objective function and the gradient of the smooth term are needed at each iteration. We show that each cluster of the iterates generated by this algorithm is a stationary point of F. Although we mainly consider the ℓ_1-regularizer in this study, the ℓ_2-norm regularization problem and the matrix trace norm problems can also be readily included in our framework, thereby broadening the capability of the algorithm. We implement the algorithm to solve problem (1.1), where f is a nonconvex smooth function from the CUTEr library to show the efficiency of the algorithm. We also run the algorithm to solve ℓ_1-regularized least squares and do performance comparisons with state-of-the-art algorithms namely NESTA, CGD, TwIST, FPC_BB, FPC_AS, and GPSR. The results of the comparisons show that the proposed algorithm is effective, comparable, and promising.

We organize the rest of this paper as follows. In Sect. 2, we briefly recall some preliminary results in optimization literature to describe our motivation for our work, construct the search direction, and present the steps of our algorithm along with some remarks. In Sect. 3, we establish the global convergence theorem under some mild conditions. In Sect. 4, we show how to extend the algorithm to solve ℓ_2-norm and matrix trace norm minimization problems. In Sect. 5, we present experiments to show the efficiency of the algorithm in solving the ℓ_1-regularized nonconvex problem and least squares problem. In Sect. 6, we conclude our paper.

2 Algorithm

2.1 Preliminary Results

First, consider the minimization of the smooth function without the ℓ_1-norm regularization

$$\min_{x\in\mathbb{R}} f(x). \tag{2.1}$$

The basic idea of Newton's method for this problem is to iteratively use the quadratic approximation q_k to the objective function $f(x)$ at the current iterate x_k and to minimize the

approximation q_k. Let $f : \mathbb{R}^n \to \mathbb{R}$ be twice continuously differentiable, and let its Hessian $G_k = \nabla^2 f(x_k)$ be positive definite. Function f at the current x_k is modeled by the quadratic approximation q_k as

$$f(x_k + s) \approx q_k(s) = f(x_k) + \nabla f(x_k)^\top s + \frac{1}{2} s^\top G_k s,$$

where $s = x - x_k$. Minimizing $q_k(s)$ yields

$$x_{k+1} = x_k - G_k^{-1} \nabla f(x_k),$$

which is Newton's formula, and $s_k = x_{k+1} - x_k = -G_k^{-1} \nabla f(x_k)$ is the so-called Newton's direction.

For the positive definite quadratic function, Newton's method can reach the minimizer with one iteration. However, when the starting point is far away from the solution, the method cannot guarantee that G_k is positive definite and that Newton's direction d_k is a descent direction. Let the quadratic model of f at x_{k+1} be

$$f(x) \approx f(x_{k+1}) + \nabla f(x_{k+1})^\top (x - x_{k+1}) + \frac{1}{2}(x - x_{k+1})^\top G_{k+1}(x - x_{k+1}).$$

Finding the derivative yields

$$\nabla f(x) \approx \nabla f(x_{k+1}) + G_{k+1}(x - x_{k+1}).$$

Setting $x = x_k$, $s_k = x_{k+1} - x_k$, and $y_k = \nabla f(x_{k+1}) - \nabla f(x_k)$, we obtain

$$G_{k+1} s_k \approx y_k. \tag{2.2}$$

For various practical problems, either the computing efforts of the Hessian matrices are very expensive or the evaluation of the Hessian is difficult; the Hessian is not even available analytically. These challenges lead to the quasi-Newton method, which generates a series of Hessian approximations through the use of the gradient while maintaining a fast rate of convergence. Instead of computing the Hessian G_k, the quasi-Newton method constructs the Hessian approximation B_k, where the sequence $\{B_k\}$ possesses positive definiteness and satisfies

$$B_{k+1} s_k = y_k. \tag{2.3}$$

In general, B_{k+1} is obtained by updating B_k with some typical and popular formulae, such as BFGS, DFP, and SR1.

Unfortunately, the standard quasi-Newton algorithm, or even its limited memory versions, does not scale well enough to train very large-scale models involving millions of variables and training instances, which are commonly encountered, for example, in natural language processing. The main computational burden of the Newton-type algorithm is the storage of a large matrix at per-iteration, which may exceed the memory capability of a personal computer (PC). Hence, a matrix-free algorithm that particularly deals with large-scale problems is urgently needed. For this purpose, the approximation Hessian B_k should be furthermore simplified as a diagonal matrix with positive components, i.e., $B_k = \lambda_k I$ with an identity matrix I and $\lambda_k > 0$. Then, the quasi-Newton condition changes to the form

$$\lambda_{k+1} I s_k = y_k.$$

Multiplying both sides by $s_k^\top$ yields

$$\lambda_{k+1}^{(1)} = \frac{s_k^\top y_k}{\|s_k\|_2^2}. \tag{2.4}$$

Similarly, multiplying both sides by $y_k^\top$, yields

$$\lambda_{k+1}^{(2)} = \frac{\|y_k\|_2^2}{s_k^\top y_k}. \tag{2.5}$$

As indicated by both formulae, if $s_k^\top y_k > 0$, then the matrix $\lambda_{k+1} I$ is positive definite, which ensures that the search direction $-\lambda_k^{-1}\nabla f(x_k)$ is descent at current point.

The formulae (2.4) and (2.5) were first developed by Barzilai and Borwein [2] for the quadratic case of f. This method essentially comprises the steepest descent method and adopts either (2.4) or (2.5) as the stepsize along a negative gradient direction. Barzilai and Borwein [2] showed that the corresponding iterative algorithm is R-superlinearly convergent for the quadratic case. Raydan [32] presented a globalization strategy based on nonmonotone line search [22] for the general non-quadratic case. Other developments of Barzilai–Borwein gradient algorithm can be found in [5,13,15–17,31,49].

2.2 Algorithm

Given its simplicity and numerical efficiency, the Barzilai–Borwein gradient method is very effective in dealing with large-scale smooth unconstrained minimization problems. However, the application of the Barilai-Borwein gradient algorithm in ℓ_1-regularized nonsmooth optimization is problematic because the regularization is non-differentiable. In this subsection, we construct an iterative algorithm to solve the ℓ_1-regularized structured nonconvex optimization problem. The algorithm can be described as the iterative form

$$x_{k+1} = x_k + \alpha_k d_k,$$

where α_k is a stepsize, and d_k is a search direction defined by minimizing a quadratic-approximated model of F.

Now, we turn our attention to consider the original problem with ℓ_1-regularizer. As ℓ_1-term is non-differentiable, at $x_k + d_k$, we use the approximated form

$$\|x_k + d_k\|_1 \approx \|x_k\| + \frac{\|x_k + hd_k\|_1 - \|x_k\|}{h},$$

where h is a small positive number, and the case $h = 1$ is reduced to the equivalent form. Subsequently, the objective function F is approximated by the quadratic approximation Q_k,

$$\begin{aligned} F(x_k + d) &= f(x_k + d) + \mu\|x_k + d\|_1 \\ &\approx f(x_k) + \nabla f(x_k)^\top d + \frac{\lambda_k}{2}\|d\|_2^2 \\ &\quad + \mu\Big[\|x_k\|_1 + \frac{\|x_k + hd\|_1 - \|x_k\|_1}{h}\Big] =: Q_k(d). \end{aligned} \tag{2.6}$$

Minimizing (2.6) yields

$$\begin{aligned} &\min_{d\in\mathbb{R}^n} Q_k(d) \\ &\Leftrightarrow \min_{d\in\mathbb{R}^n} \nabla f(x_k)^\top d + \frac{\lambda_k}{2}\|d\|_2^2 + \frac{\mu}{h}\|x_k + hd\|_1 \\ &\Leftrightarrow \min_{d\in\mathbb{R}^n} \frac{h^2}{\lambda_k}\Big(\nabla f(x_k)^\top d + \frac{\lambda_k}{2}\|d\|_2^2 + \frac{\mu}{h}\|x_k + hd\|_1\Big) \\ &\Leftrightarrow \min_{d\in\mathbb{R}^n} \frac{1}{2}\Big\|x_k + hd - \Big(x_k - \frac{h}{\lambda_k}\nabla f(x_k)\Big)\Big\|_2^2 + \frac{\mu h}{\lambda_k}\|x_k + hd\|_1 \end{aligned}$$

$$\Leftrightarrow \min_{d\in\mathbb{R}^n} \sum_{i=1}^{n} \left\{ \frac{1}{2}\left(x_k^i + hd^i - \left(x_k^i - \frac{h}{\lambda_k}\nabla f^i(x_k)\right)\right)^2 + \frac{\mu h}{\lambda_k}|x_k^i + hd^i| \right\}, \tag{2.7}$$

where x_k^i, d^i, and $\nabla f^i(x_k)$ denote the ith component of x_k, d, and $\nabla f(x_k)$, respectively. The favorable structure of (2.7) adopts the explicit solution

$$x_k^i + hd_k^i = \max\left\{ \left| x_k^i - \frac{h}{\lambda_k}\nabla f^i(x_k^i)\right| - \frac{\mu h}{\lambda_k}, 0 \right\} \frac{x_k^i - \frac{h}{\lambda_k}\nabla f^i(x_k)}{|x_k^i - \frac{h}{\lambda_k}\nabla f^i(x_k)|}.$$

Hence, the search direction at current point is

$$d_k = -\frac{1}{h}\left[x_k - \max\left\{ \left| x_k - \frac{h}{\lambda_k}\nabla f(x_k)\right| - \frac{\mu h}{\lambda_k}, 0 \right\} \frac{x_k - \frac{h}{\lambda_k}\nabla f(x_k)}{|x_k - \frac{h}{\lambda_k}\nabla f(x_k)|} \right], \tag{2.8}$$

where $|\cdot|$ and "max" are interpreted as componentwise, and the convention $0 \cdot 0/0 = 0$ is followed. When $\mu = 0$, (2.8) is reduced to $d_k = -\lambda_k^{-1}\nabla f(x_k)$, i.e., the traditional Barzilai–Borwein gradient algorithm in smooth optimization. The key motivation for this formulation is that the optimization problem in Eq. (2.7) can be easily solved by exploiting the structure of the ℓ_1-norm.

For $y \in \mathbb{R}^n$ and $\tau \in \mathbb{R}$, the unique minimizer of $\tau\|x\|_1 + \frac{1}{2}\|x - y\|^2$ is given explicitly by

$$x^* = \mathcal{S}(y, \tau) = \max\{|y| - \tau, 0\}\frac{y}{|y|}.$$

As a result, in the case of $h = 1$, (2.8) is reduced to

$$d_k = \mathcal{S}\left(x_k - \frac{1}{\lambda_k}\nabla f(x_k), \frac{\mu}{\lambda_k}\right) - x_k,$$

which is essentially the search direction of the algorithm FPC_AS [39,40]. Just because of this, in the following, we only consider the case where $h \in (0, 1)$. The important observation illustrates that our approach generalizes the definition of the search direction of FPC_AS using a small scalar h. For the reason, we believe that the search direction should be generated by theoretically minimizing the quadratic approximation of both terms in F, not just in the smooth function f. Another advantage of our approach is that an appropriate value of h may result in an improved numerical performance, and that the approach is not restricted to the special case $h = 1$.

Lemma 2.1 *For any real vectors $a \in \mathbb{R}^n$ and $b \in \mathbb{R}^n$, the following function $L(x)$ is non-decreasing*

$$L(x) = \frac{\|a + bx\|_1 - \|a\|_1}{x}, \quad x \in (0, \infty). \tag{2.9}$$

Proof Note that

$$L(x) = \frac{\|a + bx\|_1 - \|a\|_1}{x} = \sum_{i}^{n} \frac{|a^i + b^i x| - |a^i|}{x} \triangleq \sum_{i}^{n} l^i(x);$$

hence, it is reduced to prove that $l^i(x)$ is non-decreasing for each i.

(a) When $a^i \geq 0$ and $a^i x + b^i \geq 0$, $l^i(x) = b^i$.

(b) When $a^i \geq 0$ and $a^i x + b^i \leq 0$, we obtain

$$l^i(x) = \frac{-2a^i - b^i x}{x} = \frac{-2a^i}{x} - b^i.$$

(c) When $a^i \leq 0$ and $a^i x + b^i \geq 0$, we obtain

$$l^i(x) = \frac{2a^i + b^i x}{x} = \frac{2a^i}{x} + b^i.$$

(d) When $a^i \leq 0$ and $a^i x + b^i \geq 0$, we have $l^i(x) = -b^i$.

Clearly $l^i(x)$ is non-decreasing for each case. Hence, $L(x)$ is non-decreasing. □

The following lemma shows that the direction defined by (2.8) is descent if $d_k \neq 0$.

Lemma 2.2 *Suppose that $\lambda_k > 0$ and d_k is determined by (2.8). Then,*

$$F(x_k + \theta d_k) \leq F(x_k) + \theta\left[\nabla f(x_k)^\top d_k + \frac{\mu\|x_k + h d_k\|_1 - \mu\|x_k\|_1}{h}\right] + o(\theta) \quad \theta \in (0, h], \tag{2.10}$$

and

$$\nabla f(x_k)^\top d_k + \frac{\mu\|x_k + h d_k\|_1 - \mu\|x_k\|_1}{h} \leq -\frac{\lambda_k}{2}\|d_k\|_2^2. \tag{2.11}$$

Proof By the differentiability of f and the convexity of $\|x\|_1$, we show that for any $\theta \in (0, h]$ $(\theta/h \in (0, 1])$,

$$\begin{aligned}
F(x_k + \theta d_k) - F(x_k) &= f(x_k + \theta d_k) - f(x_k) + \mu\|x_k + \theta d_k\|_1 - \mu\|x_k\|_1 \\
&= f(x_k + \theta d_k) - f(x_k) + \mu\left\|\frac{\theta}{h}(x_k + h d_k) + \left(1 - \frac{\theta}{h}\right)x_k\right\|_1 - \mu\|x_k\|_1 \\
&\leq f(x_k + \theta d_k) - f(x_k) + \frac{\theta\mu}{h}\|x_k + h d_k\|_1 + \left(1 - \frac{\theta}{h}\right)\mu\|x_k\|_1 - \mu\|x_k\|_1 \\
&= \theta\nabla f(x_k)^\top d_k + o(\theta) + \theta\left[\frac{\mu}{h}\|x_k + h d_k\|_1 - \frac{\mu}{h}\|x_k\|_1\right],
\end{aligned}$$

which is exactly (2.10).

Noting d_k is the minimizer of (2.6) and $\theta \in (0, h]$, from (2.6) as well as the convexity of $\|x\|_1$, we have

$$\begin{aligned}
&\nabla f(x_k)^\top d_k + \frac{\lambda_k}{2}\|d_k\|_2^2 + \frac{\mu\|x_k + h d_k\|_1 - \mu\|x_k\|_1}{h} \\
&\quad \leq \theta\nabla f(x_k)^\top d_k + \frac{\lambda_k}{2}\|\theta d_k\|_2^2 + \frac{\mu}{h}\|x_k + \theta h d_k\|_1 - \frac{\mu}{h}\|x_k\|_1 \\
&\quad \leq \theta\nabla f(x_k)^\top d_k + \frac{\lambda_k\theta^2}{2}\|d_k\|_2^2 + \frac{\theta\mu}{h^2}\|x_k + h^2 d_k\|_1 + \frac{\mu}{h}\left(1 - \frac{\theta}{h}\right)\|x_k\|_1 - \frac{\mu}{h}\|x_k\|_1.
\end{aligned}$$

Hence,

$$\begin{aligned}
&(1 - \theta)\nabla f(x_k)^\top d_k + \frac{\mu}{h}\|x_k + h d_k\|_1 - \frac{\theta\mu}{h^2}\|x_k + h^2 d_k\|_1 \\
&\quad - \frac{\mu}{h}\left(1 - \frac{\theta}{h}\right)\|x_k\|_1 \leq -\frac{\lambda_k}{2}\left(1 - \theta^2\right)\|d_k\|_2^2.
\end{aligned} \tag{2.12}$$

The last three terms of the left hand side in (2.12) can be re-organized as

$$\begin{aligned}
&\frac{\mu}{h}\left\{\|x_k+hd_k\|_1-\frac{\theta}{h}\|x_k+h^2d_k\|_1-\left(1-\frac{\theta}{h}\right)\|x_k\|_1\right\}\\
&=\frac{\mu}{h}\left\{\|x_k+hd_k\|_1-\|x_k\|_1-\theta\left[\frac{\|x_k+h^2d_k\|_1-\|x_k\|_1}{h}\right]\right\}\\
&=\frac{\mu}{h}\left\{\|x_k+hd_k\|_1-\|x_k\|_1-\theta\left[h\cdot\frac{\|x_k+h^2d_k\|_1-\|x_k\|_1}{h^2}\right]\right\}\\
&\geq\frac{\mu}{h}\left\{\|x_k+hd_k\|_1-\|x_k\|_1-\theta\left[h\cdot\frac{\|x_k+hd_k\|_1-\|x_k\|_1}{h}\right]\right\}\\
&=\frac{\mu}{h}(1-\theta)\left\{\|x_k+hd_k\|_1-\|x_k\|_1\right\},
\end{aligned}\tag{2.13}$$

where the inequality is from Lemma 2.1. Combining (2.12) with (2.13), we obtain

$$(1-\theta)\nabla f(x_k)^\top d_k+(1-\theta)\frac{\mu\|x_k+hd_k\|_1-\mu\|x_k\|_1}{h}\leq-\frac{\lambda_k}{2}\left(1-\theta^2\right)\|d_k\|_2^2.\tag{2.14}$$

Dividing both sides of (2.14) by $(1-\theta)$ and noting $1-\theta>0$ when $h\in(0,1)$, we arrive at the desired result (2.11). □

When the search direction is determined, a suitable stepsize along this direction should be found to determine the next iterative point. In this study we pay particular attention to a nonmonotone line search strategy, which differs from the traditional Armijo line search or the Wolfe–Powell line search. The traditional Armijo line search requires the function value to decrease monotonically at each iteration. This requirement may cause the sequence of iterations to follow the bottom of a curved narrow valley, which commonly occurs in difficult nonlinear problems. To overcome this difficulty, a possible alternative is to allow an occasional increase in the objective function at each iteration. To clarify further the proposed algorithm, we briefly recall the earliest nonmonotone line search technique by Grippo, Lampariello, and Lucidi [22]. Let $\delta\in(0,1)$, $\rho\in(0,1)$, and $\tilde{m}$ be a positive integer. The nonmonotone line search is used to choose the smallest nonnegative integer j_k such that the stepsize $\alpha_k=\tilde{\alpha}\rho^{j_k}$ satisfies

$$f(x_k+\alpha_kd_k)\leq\max_{0\leq j\leq m(k)}f(x_{k-j})+\delta\alpha_k\nabla f(x_k)^\top d_k,\tag{2.15}$$

where

$$m(0)=0\quad\text{and}\quad0\leq m(k)\leq\min\{m(k-1)+1,\tilde{m}\}.$$

If $m(k)=0$, the above nonmonotone line search reduces to the standard Armijo line search.

Based on Lemma 2.2, the inequality (2.15) should be modified as

$$F(x_k+\alpha_kd_k)\leq\max_{0\leq j\leq m(k)}F(x_{k-j})+\delta\alpha_k\Delta_k,\tag{2.16}$$

where

$$\Delta_k=\nabla f(x_k)^\top d_k+\frac{\mu\|x_k+hd_k\|_1-\mu\|x_k\|_1}{h}.\tag{2.17}$$

As shown in (2.11) $\Delta_k\leq-\frac{\lambda_k}{2}\|d_k\|_2^2<0$ whenever $d_k\neq0$. Hence, α_k given by (2.16) is well-defined.

Given all the derivations above, we now describe the nonmonotone Barzilai–Borwein gradient algorithm (hereafter referred to NBBL1) as follows.

ALGORITHM 1 **(NBBL1)**

Initialization: *Choose* x_0 *and constant* $\mu > 0$. *Constants* $\tilde{\alpha} > 0$, $\rho \in (0, 1)$, $\delta \in (0, 1)$, $h \in (0, 1]$ *and positive integer* $\tilde{m}$. *Set* $k = 0$.
Step 1. *Stop if* $\|d_k\|_2 = 0$. *Otherwise, continue.*
Step 2. *Compute* d_k *via (2.8).*
Step 3. *Compute* α_k *via (2.16).*
Step 4. *Let* $x_{k+1} = x_k + \alpha_k d_k$.
Step 5. *Let* $k = k + 1$. *Go to Step 1.*

Remark 1 If $\lambda_k > 0$, then the generated direction is descent. However, the condition $\lambda_k > 0$ in this case may not be fulfilled and the hereditary descent property can no longer be guaranteed. To cope with this drawback, we should keep the sequence $\{\lambda_k\}$ uniformly bounded; that is, for sufficiently small $\lambda_{(\min)} > 0$ and sufficiently large $\lambda_{(\max)} > 0$, the λ_k is forced as

$$\lambda_k = \min\left\{\lambda_{(\max)}, \max\left\{\lambda_k, \lambda_{(\min)}\right\}\right\}.$$

This approach ensures that λ_k is bounded from zero and subsequently ensures that d_k is descent at per-iteration.

Remark 2 Lemma 2.2demonstrates the existence of a constant $\theta \in (0, h]$ such that $x_k + \theta d_k$ is a descent point. Hence, choosing the initial stepsize as $\tilde{\alpha} = h$ is suggested in practical computation.

3 Convergence Analysis

Thfis section is devoted to presenting some favorable properties of the generated direction and establishing the global convergence of Algorithm 1. Our convergence result utilizes the following assumption:

Assumption 1 The level set $\Omega = \{x : f(x) \leq f(x_0)\}$ is bounded.

To easily understand the lemma given below, we present two frequently used definitions in convex analysis.

Definition 3.1 The directional derivative of a multivariate function $f(x_1, \ldots, x_n)$ along a given vector $d = (d_1, \ldots, d_n)$ at a given point x is defined by

$$f'(x; d) = \lim_{\alpha \downarrow 0} \frac{f(x + \alpha d) - f(x)}{\alpha}.$$

Definition 3.2 A feasible point $x \in \mathbb{R}^n$ is said to be a stationary point of f if $f'(x; d) \geq 0$ for all $d \in \mathbb{R}^n$.

Lemma 3.3 *Suppose that* $\lambda_k > 0$ *and* d_k *is defined by (2.8) with* $h \in (0, 1]$. *Then,* x_k *is a stationary point of problem (1.1) if and only if* $d_k = 0$.

Proof If $d_k \neq 0$, then Lemma 2.2 shows that d_k is descent direction at x_k, which implies that x_k is not a stationary point of F. If $d_k = 0$ is the solution of (2.7), we obtain the following for any $\alpha d \in \mathbb{R}^n$ with $\alpha > 0$:

$$\alpha \nabla f(x_k)^\top d + \frac{\lambda_k \alpha^2}{2}\|d\|_2^2 + \frac{\mu}{h}\|x_k + \alpha h d\|_1 \geq \frac{\mu}{h}\|x_k\|_1. \tag{3.1}$$

Combining $f(x_k + \alpha d) - f(x_k) = \alpha \nabla f(x_k)^\top d + o(\alpha)$ with (3.1) yields

$$\begin{aligned}
F'(x_k; d) &= \lim_{\alpha \downarrow 0} \frac{f(x_k + \alpha d) - f(x_k) + \mu\|x_k + \alpha d\|_1 - \mu\|x_k\|_1}{\alpha} \\
&= \lim_{\alpha \downarrow 0} \frac{\alpha \nabla f(x_k)^\top d + o(\alpha) + \mu\|x_k + \alpha d\|_1 - \mu\|x_k\|_1}{\alpha} \\
&\geq \lim_{\alpha \downarrow 0} \Bigg(\frac{-\frac{\lambda_k \alpha^2}{2}\|d\|_2^2 + o(\alpha)}{\alpha} \\
&\quad + \frac{\left[\mu\|x_k + \alpha d\|_1 - \mu\|x_k\|_1\right] - \left[\frac{\mu}{h}\|x_k + \alpha h d\|_1 - \frac{\mu}{h}\|x_k\|_1\right]}{\alpha} \Bigg). \\
&\geq \lim_{\alpha \downarrow 0} \frac{-\frac{\lambda_k \alpha^2}{2}\|d\|_2^2 + o(\alpha)}{\alpha} \\
&= 0,
\end{aligned}$$

where the second inequality is from Lemma 2.1. Hence, x_k is a stationary point of F. □

The proof of the following lemma is similar to the Theorem in [22].

Lemma 3.4 *Let $l(k)$ be an integer such that*

$$k - m(k) \leq l(k) \leq k \quad \text{and} \quad F(x_{l(k)}) = \max_{0 \leq j \leq m(k)} F(x_{k-j}).$$

Then, the sequence $\{F(x_{l(k)})\}$ is non-increasing, and the search direction $d_{l(k)}$ satisfies

$$\lim_{k \to \infty} \alpha_{l(k)} \|d_{l(k)}\|_2 = 0. \tag{3.2}$$

Proof From the definition of $m(k)$, we have $m(k+1) \leq m(k) + 1$. Hence,

$$\begin{aligned}
F(x_{l(k+1)}) &= \max_{0 \leq j \leq m(k+1)} F(x_{k+1-j}) \\
&\leq \max_{0 \leq j \leq m(k)+1} F(x_{k+1-j}) \\
&= \max\left\{F(x_{l(k)}), F(x_{k+1})\right\} \\
&= F(x_{l(k)}).
\end{aligned}$$

Based on (2.16), we obtain the following for all $k > \tilde{m}$:

$$\begin{aligned}
F(x_{l(k)}) &= F\left(x_{l(k)-1} + \alpha_{l(k)-1} d_{l(k)-1}\right) \\
&\leq \max_{0 \leq j \leq m(l(k)-1)} F\left(x_{l(k)-1-j}\right) + \delta \alpha_{l(k)-1} \Delta_{l(k)-1} \\
&= F\left(x_{l(l(k)-1)}\right) + \delta \alpha_{l(k)-1} \Delta_{l(k)-1}.
\end{aligned}$$

By assumption 1, the sequence $\{F(x_{l(k)})\}$ admits a limit for $k \to \infty$. Hence,

$$\lim_{k \to \infty} \alpha_{l(k)} \Delta_{l(k)} = 0. \tag{3.3}$$

Given the definition of Δ_k in (2.17) and the inequality (2.11), we can deduce that

$$\Delta_{l(k)} \leq -\frac{\lambda_{(\min)}}{2}\|d_{l(k)}\|_2^2 < 0.$$

Combining the previous equation with (3.3) yields

$$\lim_{k\to\infty} \alpha_{l(k)}\|d_{l(k)}\|_2^2 = 0,$$

which shows the desired result (3.2). □

Theorem 3.5 *Let the sequences $\{x_k\}$ and $\{d_k\}$ be generated by Algorithm 1. Then, a subsequence $\mathcal{K}$ exists such that*

$$\lim_{k\to\infty, k\in\mathcal{K}} \|d_k\|_2 = 0. \tag{3.4}$$

Proof As shown in [22], (3.2) also implies that

$$\lim_{k\to\infty} \alpha_k\|d_k\|_2 = 0. \tag{3.5}$$

Now, let $\bar{x}$ be a limit point of $\{x_k\}$ and $\{x_k\}_{\mathcal{K}_1}$ be a subsequence of $\{x_k\}$ converging to $\bar{x}$. Then, by (3.5), either $\lim_{k\to\infty, k\in\mathcal{K}_1} \|d_k\|_2 = 0$, which implies $\|\bar{d}\|_2 = 0$, or a subsequence $\{x_k\}_{\mathcal{K}}$ ($\mathcal{K} \subset \mathcal{K}_1$) exists such that

$$\lim_{k\to\infty, k\in\mathcal{K}} d_k \neq 0 \quad \text{and} \quad \lim_{k\to\infty, k\in\mathcal{K}} \alpha_k = 0. \tag{3.6}$$

In this case, we assume that a constant $\epsilon > 0$ exists such that

$$\|d_k\|_2 \geq \epsilon, \quad \forall\, k \in \mathcal{K}. \tag{3.7}$$

As α_k is the first value that satisfies (2.16), we can assume based on Step 3 in Algorithm 1 that an index $\bar{k}$ exists such that for all $k \geq \bar{k}$ and $k \in \mathcal{K}$,

$$F\left(x_k + \frac{\alpha_k}{\rho}d_k\right) > \max_{0\leq j\leq m(k)} F(x_{k-j}) + \delta\frac{\alpha_k}{\rho}\Delta_k \geq F(x_k) + \delta\frac{\alpha_k}{\rho}\Delta_k. \tag{3.8}$$

As f is continuously differentiable, we can find based on the mean-value theorem on f that a constant $\theta_k \in (0, 1)$ exists such that

$$f\left(x_k + \frac{\alpha_k}{\rho}d_k\right) - f(x_k) = \frac{\alpha_k}{\rho}\nabla f\left(x_k + \theta_k\frac{\alpha_k}{\rho}d_k\right)^\top d_k.$$

By combining the previous equation with (3.8), we obtain

$$\nabla f\left(x_k + \theta_k\frac{\alpha_k}{\rho}d_k\right)^\top d_k + \frac{\mu\|x_k + \frac{\alpha_k}{\rho}d_k\|_1 - \mu\|x_k\|_1}{\alpha_k/\rho} > \delta\Delta_k. \tag{3.9}$$

As $\tilde{\alpha} = h$ and $\alpha_k \to 0$ in (3.6), we obtain $\alpha_k < \rho h$ as $k \to \infty$. Based on Lemma 2.1,

$$\frac{\mu\|x_k + \frac{\alpha_k}{\rho}d_k\|_1 - \mu\|x_k\|_1}{\alpha_k/\rho} - \frac{\mu\|x_k + hd_k\|_1 - \mu\|x_k\|_1}{h} \leq 0.$$

Subtracting both sides of (3.9) by Δ_k and noting the definition of Δ_k

$$\nabla f\left(x_k + \theta_k\frac{\alpha_k}{\rho}d_k\right)^\top d_k - \nabla f(x_k)^\top d_k$$

$$\begin{aligned}
&\geq \nabla f\left(x_k + \theta_k \frac{\alpha_k}{\rho} d_k\right)^\top d_k - \nabla f(x_k)^\top d_k \\
&\quad + \left[\frac{\mu\|x_k + \frac{\alpha_k}{\rho} d_k\|_1 - \mu\|x_k\|_1}{\alpha_k/\rho} - \frac{\mu\|x_k + h d_k\|_1 - \mu\|x_k\|_1}{h}\right] \\
&> -(1-\delta)\Delta_k \\
&\geq (1-\delta)\frac{\lambda_{(\min)}}{2}\|d_k\|_2^2. \qquad (3.10)
\end{aligned}$$

Taking the limit as $k \in \mathcal{K}$, $k \to \infty$ in both sides of (3.10) and using the smoothness of f, we obtain

$$0 = \nabla f(\bar{x})^\top \bar{d} - \nabla f(\bar{x})^\top \bar{d} \geq (1-\delta)\frac{\lambda_{(\min)}}{2}\|\bar{d}\|_2^2,$$

which implies that $\|d_k\|_2 \to 0$ as $k \in \mathcal{K}$, $k \to \infty$. This result yields a contradiction because (3.7) indicates that $\|d_k\|_2$ is bounded away from zero. □

4 Some Extensions

In this section, we show that our algorithm can be readily extended to solve ℓ_2-norm and matrix trace norm minimization problems in machine learning, thereby broadening the applicable range of our approach significantly.

First, we consider the ℓ_2-regularization problem

$$\min_{x\in\mathbb{R}^n} \; F(x) = f(x) + \mu\|x\|_2.$$

The search direction d_k is clearly determined by minimizing

$$\min_{d\in\mathbb{R}^n} \frac{1}{2}\left\|x_k + hd - \left(x_k - \frac{h}{\lambda_k}\nabla f(x_k)\right)\right\|_2^2 + \frac{\mu h}{\lambda_k}\|x_k + hd\|_2.$$

Based on [19], the explicit solution is

$$x_k + hd_k = \max\left\{\left\|x_k - \frac{h}{\lambda_k}\nabla f(x_k)\right\|_2 - \frac{\mu h}{\lambda_k}, 0\right\}\frac{x_k - \frac{h}{\lambda_k}\nabla f(x_k)}{\|x_k - \frac{h}{\lambda_k}\nabla f(x_k)\|_2},$$

i.e.,

$$d_k = -\frac{1}{h}\left[x_k - \max\left\{\left\|x_k - \frac{h}{\lambda_k}\nabla f(x_k)\right\|_2 - \frac{\mu h}{\lambda_k}, 0\right\}\frac{x_k - \frac{h}{\lambda_k}\nabla f(x_k)}{\|x_k - \frac{h}{\lambda_k}\nabla f(x_k)\|_2}\right].$$

Now, we consider the matrix trace norm minimization problem

$$\min_{X\in\mathbb{R}^{m\times n}} \; F(X) = f(X) + \mu\|X\|_*, \qquad (4.1)$$

where the functional $\|X\|_*$ is the trace norm of matrix X, which is defined as the sum of its singular values. That is, assume that X has r positive singular values of $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r \geq 0$; then, $\|X\|_* = \sum_{i=1}^r \sigma_i$. The matrix trace norm is also known as the Schatten ℓ_1-norm, Ky Fan norm, and nuclear norm [33]. This problem has received much attention because it is closely related to the affine rank minimization problem, which has appeared in many control applications, including controller design, realization theory, and model reduction.

Similar to the steps undertaken in previous sections, we can readily reformulate (2.6) as the following quadratic model to determine the search direction:

$$\min_{D\in\mathbb{R}^{m\times n}} \frac{1}{2}\left\| X_k + hD - \left(X_k - \frac{h}{\lambda_k}\nabla f(X_k)\right)\right\|_2^2 + \frac{\mu h}{\lambda_k}\|X_k + hD\|_*. \tag{4.2}$$

To obtain the exact solution of (4.2), we now consider the singular value decomposition (SVD) of a matrix $Y \in \mathbb{R}^{m\times n}$ with rank r as

$$Y = U\Sigma V^\top, \quad \Sigma = \mathrm{diag}(\{\sigma_i\}_{1\le i\le r}),$$

where U and V are $m \times r$ and $r \times n$ matrices, respectively, with orthonormal columns, and the singular value σ_i is positive. For each $\tau > 0$, let

$$\mathcal{D}_\tau(Y) = U\mathcal{D}_\tau(\Sigma)V^\top, \quad \mathcal{D}_\tau(\Sigma) = \mathrm{diag}([\sigma_i - \tau]_+),$$

where $[\cdot]_+ = \max\{0, \cdot\}$. $\mathcal{D}_\tau(Y)$ obeys the following nuclear norm minimization problem [7], i.e.,

$$\mathcal{D}_\tau(Y) = \arg\min_X \tau\|X\|_* + \frac{1}{2}\|X - Y\|_F^2. \tag{4.3}$$

Comparing (4.2) with (4.3), we deduce that

$$X_k + hD_k = U\mathcal{D}_{\mu h/\lambda_k}(\Sigma)V^\top \quad \text{and} \quad \mathcal{D}_{\mu h/\lambda_k}(\Sigma) = \mathrm{diag}\big([\sigma_i - \frac{\mu h}{\lambda_k}]_+\big),$$

or, equivalently,

$$D_k = -\frac{1}{h}\Big[X_k - U\mathcal{D}_{\mu h/\lambda_k}(\Sigma)V^\top\Big].$$

Subsequently, the NBBL1 framework to solve ℓ_2-norm and matrix trace norm regularization problems is easily derived.

5 Numerical Experiments

In this section, we present numerical results to illustrate the feasibility and efficiency of NBBL1. We partition our experiments into three classes based on different types of f. In the first class, we use our algorithm to solve ℓ_1-regularized nonconvex problem. In the second class, we use our algorithm to solve ℓ_1-regularized least squares, which mainly appear in compressive sensing. In the third class, we compare some state-of-the-art algorithms in compressive sensing to show the efficiency of our algorithm. All experiments are performed under Windows XP and Matlab 7.8 (2009a) running on a Lenovo laptop with an Intel Atom CPU at 1.6 GHz and 1 GB of memory.

5.1 Test on ℓ_1-Regularized Nonconvex Problem

Our first test is performed on a set of nonconvex unconstrained problems from the CUTEr [14] library. The second-order derivatives of all the selected problems are available. Given our interest in large problems, we only consider the problems with a size of at least 100. For such problems, we use the dimensions that is admissible of the "double large" installation of CUTEr. The algorithm stops if the norm of the search direction is small enough; that is,

$$\|d_k\|_2 \le tol_1. \tag{5.1}$$

Table 1 Test result for NBBL1 with $\mu = 0$

Problem	Dim	μ	Iter	Nf	Time	Fun	Normg	Normd
VARDIM	1,000	0.0	49	94	0.48	3.2506e−26	3.6059e−13	2.5893e−09
FLETCHER	100	0.0	1,217	1,983	3.75	3.0113e−10	2.2576e−05	9.9505e−09
COSINE	10,000	0.0	51	350	23.41	−9.9990e+03	2.5387e−03	4.4188e−09
SINQUAD	1,000	0.0	180	908	10.22	6.4479e−05	4.9743e−05	6.8482e−09
GENROSE	200	0.0	323	646	0.80	1.0000e+00	1.3870e−05	9.9488e−09
WOODS	1,000	0.0	322	579	3.00	9.9104e−13	7.2693e−06	7.5155e−09
NONCVXU2	200	0.0	4,987	8,476	21.17	4.6373e+02	2.0726e−07	7.0300e−09
BROYDN7D	500	0.0	1,305	2,402	10.38	3.8234e+00	9.3966e−07	9.2037e−09
CHAIWOO	1,000	0.0	757	1,335	9.80	1.0000e+00	8.0006e−06	4.4747e−09

The iterative process is also stopped if the number of iterations exceeds 10,000 without achieving convergence.

In this experiment, we take $tol_1 = 10^{-8}$, $h = 1$, $\lambda_{(\min)} = 10^{-20}$, and $\lambda_{(\max)} = 10^{20}$. In the line search, we choose $\tilde{\alpha}_0 = 1$, $\rho = 0.35$, $\delta = 10^{-4}$, and $\tilde{m} = 5$. We test NBBL1 with different parameter values $\mu = \{0, .25, 1, 1.5, 2\}$. The numerical results are presented in Tables 1 and 2, which contain the name of the problem (Problem), the dimensions of the problem (Dim), the number of iterations (Iter), the number of function evaluations (Nf), the CPU time required in seconds (Time), the final objective function values (Fun), the norm of the final gradient of f (Normg), or the number of nonzeros components of solutions (Nzero), and the norm of final direction (Normd).

As show in Tables 1 and 2, NBBL1 works successfully for all the test problems in each case. Particularly, NBBL1 consistently produces highly accurate solutions within a short amount of time. The proposed algorithm requires a large number of iterations for a number of special problems, such as problems `FLETCHER`, `NONCVXU2`, and `BROYDN7D` with parameter $\mu = 0$, problems `FLETCHER` and `BROYDN7D` with $\mu = 0.25$; problems `FLETCHER` and `CHAIWOO` with $\mu = 1$, problems `VARDIM`, `FLETCHER` and `CHAIWOO` with $\mu = 1.5$ and $\mu = 2$. It can also be observed that the number of iterations and function evaluations both decrease universally as μ increase from 1 except for problem `VARDIM`. As illustrated at the third column on the right, the number of the non-zero components of final solutions decrease dramatically as μ gets small, and reach zero when $\mu = 2$ for a coupe of problems. Moreover, if larger μ is permitted, the number of non-zero components of final solutions are all zero for each test problem, which means that zero is the optimal solution. The phenomenon is not surprising once we note the separable structure of the original problem and the balanced parameter μ. From Table 2 we also note that for problems `COSINE` and `GENROSE`, the proposed algorithm requires only two steps to achieve convergence. From CUTEr [14], it is easy to see that the fundamental cause of the thing lies in the starting point near to zero, i.e. the solution. Meanwhile, the norm of the final direction is exactly zero for problems `COSINE` and `GENROSE`, because at the solution it has $x_k = 0$ and $|x_k - \frac{h}{\lambda_k}\nabla f(x_k)| < \frac{\mu h}{\lambda_k}$ in (2.8) at this case. Table 1 presents the numerical results of NBBL1 after solving a smooth nonconvex minimization problem without any regularization. As shown in the second to the last column of this table, the norm of the final gradient is sufficiently small. The important observation verifies that the proposed algorithm is very efficient in solving unconstrained smooth minimization problems. This result is not surprising, because the proposed algorithm reduces to the effective nonmonotone Barzailai-Borwein gradient method of Raydan [32] in this case.

Table 2 Test result for NBBL1 with $\mu = 0.25, 1, 2$

Problem	Dim	μ	Iter	Nf	Time	NZero	Fun	Normd
VARDIM	1,000	0.25	49	94	0.06	1,000	2.5000e+02	6.4503e−09
FLETCHER	100	0.25	5,042	8,657	1.65	100	2.4497e+01	9.7495e−09
COSINE	10,000	0.25	47	108	2.23	9,999	−1.6829e+03	4.8382e−09
SINQUAD	1,000	0.25	46	92	0.19	2	2.8084e−01	2.2567e−13
GENROSE	200	0.25	9	57	0.00	199	1.9846e+02	3.7742e−09
WOODS	1,000	0.25	645	1,527	0.80	1,000	2.4911e+02	7.0300e−09
NONCVXU2	200	0.25	998	1,957	0.58	176	5.6230e+02	8.2357e−09
BROYDN7D	500	0.25	1,314	2,366	1.56	500	8.9609e+01	8.1753e−09
CHAIWOO	1,000	0.25	435	746	0.72	1,000	2.5055e+02	6.6783e−09
VARDIM	1,000	1.0	143	450	0.16	1,000	9.3925e+02	6.2518e−09
FLETCHER	100	1.0	1,046	1,635	0.33	100	9.7761e+01	4.0598e−09
COSINE	10,000	1.0	63	983	3.06	1	1.0462e+04	7.6739e−12
SINQUAD	1,000	1.0	59	96	0.20	2	6.3115e−01	4.7160e−11
GENROSE	200	1.0	8	52	0.00	199	1.9950e+02	4.4169e−10
WOODS	1,000	1.0	837	1,826	0.94	1,000	9.8571e+02	9.5164e−09
NONCVXU2	200	1.0	492	774	0.27	140	6.2045e+02	8.7467e−09
BROYDN7D	500	1.0	510	975	0.59	494	3.3286e+02	7.5406e−09
CHAIWOO	1,000	1.0	1,057	1,804	1.62	1,000	9.9376e+02	5.0433e−09
VARDIM	1,000	1.5	845	2,780	0.94	1,000	1.3596e+03	6.0186e−09
FLETCHER	100	1.5	1,323	2,366	0.45	100	1.4645e+02	8.9985e−09
COSINE	10,000	1.5	3	4	0.19	0	9.9990e+03	0.0000e+00
SINQUAD	1,000	1.5	83	339	0.48	2	7.7024e−01	4.5581e−09
GENROSE	200	1.5	5	6	0.00	199	1.9988e+02	5.0868e−09
WOODS	1,000	1.5	853	1,809	0.83	1,000	1.4677e+03	9.9695e−09
NONCVXU2	200	1.5	365	671	0.19	107	6.7764e+02	5.7948e−09
BROYDN7D	500	1.5	305	517	0.22	26	4.9970e+02	7.7333e−09
CHAIWOO	1,000	1.5	935	1,538	1.53	1,000	1.4962e+03	8.6569e−09
VARDIM	1,000	2.0	1,506	4,816	1.64	1,000	1.7511e+03	2.4783e−09
FLETCHER	100	2.0	996	1,721	0.33	100	2.4344e+02	2.8322e−09
COSINE	10,000	2.0	2	3	0.14	0	9.9990e+03	0.0000e+00
SINQUAD	1,000	2.0	67	110	0.23	2	8.6555e−01	5.0413e−11
GENROSE	200	2.0	2	3	0.03	0	2.0000e+02	0.0000e+00
WOODS	1,000	2.0	102	283	0.11	750	3.3572e+03	3.8964e−09
NONCVXU2	200	2.0	251	825	0.14	50	7.3779e+02	9.7233e−09
BROYDN7D	500	2.0	152	455	0.12	9	5.0216e+02	5.0308e−09
CHAIWOO	1,000	2.0	927	1,634	1.54	998	1.9739e+03	9.5918e−09

5.2 Test on ℓ_1-Regularized Least Squares

Let $\bar{x}$ be a sparse or a nearly sparse original signal, $A \in \mathbb{R}^{m\times n}$ $(m \ll n)$ be a linear operator, $\omega \in \mathbb{R}^m$ be a zero-mean Gaussian white noise, and $b \in \mathbb{R}^m$ be an observation that satisfies the relationship

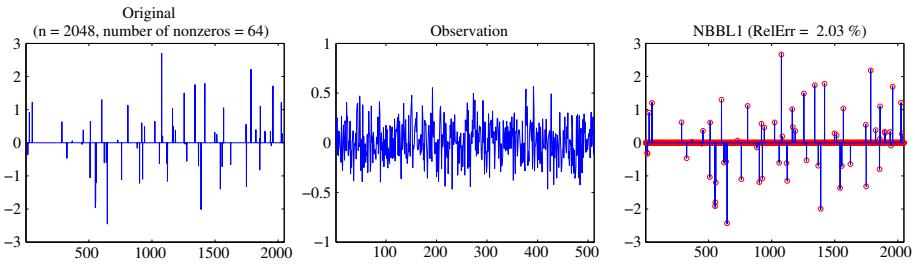

Fig. 1 *Left* Original signal with length 4,096 and 64 positive non-zero elements; *Middle* the noisy measurement with length 512; *Right* recovered signal by NBBL1 (*red circle*) versus original signal (*blue peaks*) (Color figure online)

$$b = A\bar{x} + \omega.$$

Recent compressive sensing results show that under some technical conditions, the desired signal can be reconstructed almost exactly by solving the ℓ_1-regularized least squares (1.2). In this subsection, we perform two classes of numerical experiments to solve (1.2) using the Gaussian matrices as the encoder. In the first class, we demonstrate that our algorithm effectively decodes a sparse signal. In the second class, we carry out a series of experiments with different h to choose the best one. We measure the quality of restoration x^* through the relative error to the original signal $\bar{x}$; that is,

$$\text{RelErr} = \frac{\|x^* - \bar{x}\|_2}{\|\bar{x}\|_2}. \tag{5.2}$$

In the first test, we use a random matrix A with independent identically distributed Gaussian entries. The ω is the additive Gaussian noise of zero mean and standard deviation σ. Given the storage limitations of the PC, we test a small size signal with $n = 2^{11}$, $m = 2^9$. The original signal contains randomly $p = 2^6$ nonzero elements. We also choose the noise level $\sigma = 10^{-3}$. The proposed algorithm starts at a zero point and terminates when the relative change of two successive points are sufficiently small, i.e.,

$$\frac{\|x_k - x_{k-1}\|_2}{\|x_{k-1}\|_2} < tol_2. \tag{5.3}$$

In this experiment, we take $tol = 10^{-4}$, $h = 10^{-2}$, $\lambda_{(\min)} = 10^{-30}$, and $\lambda_{(\max)} = 10^{30}$. In the line search, we choose $\tilde{\alpha}_0 = 10^{-2}$, $\rho = 0.35$, $\delta = 10^{-4}$, and $\tilde{m} = 5$. The original signal, the limited measurement, and the reconstructed signal are given in Fig. 1.

Comparing the left plot with the right one in Fig. 1, we clearly see that the original sparse signal is restored almost completely. All the blue peaks are encircled by the red circles, illustrating that the original signal has been found almost exactly. Overall, this simple experiment shows that the proposed algorithm performs quite well and provides an efficient approach to the recovery of large sparse non-negative signal.

The last term in the approximate quadratic model (2.6) is equivalent to $\|x_k + d\|_1$ exactly when $h = 1$. Next, we provide evidence to show that other values can be potentially and dramatically better than $h = 1$. We conduct a series of experiments and compare the performance at each case. In our experiments, we set all the parameters values as in the previous test except for $n = 2^{10}$. We present in Fig. 2 the impact of the parameter h values on the total number of iterations, the computing time, and the quality of the recovered signal. In each plot, the level axis denotes the values of h from 0.01 to 1 in a log scale.

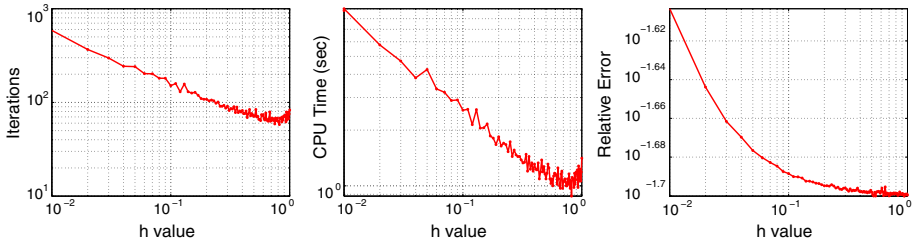

Fig. 2 Performance of NBBL1: number of iterations (*left*), computing time (*middle*) and final relative error (*right*). In each plot, the *horizontal axis* represents the value of h in log scale

In Fig. 2, the number of iterations, the computing time, and the quality of restorations are greatly influenced by the h values. Generally, as h increases, NBBL1 consistently performs satisfactorily. On the other hand, the right plot clearly demonstrates that the relative error decreases dramatically at the very beginning and then decreases slightly after 0.2. However, the quality of restoration cannot be improved further after 0.7. On the other hand, the left and middle plots show that the number of iterations and the computing time slightly increase after $h = 0.8$. The these plots verify that the performance of NBBL1 is sensitive to the h values, and that the value of $h \in [0.7, 0.9]$ may be preferable.

5.3 Comparisons with NESTA_Ct, GPSR_BB, CGD, TwIST, FPC_BB, and FPC_AS

In the third class of the experiment we test the proposed algorithm against several state-of-the-art algorithms to solve ℓ_1-regularized problems in compressive sensing or linear inverse problems. Compare each algorithm in a very fair way is difficult, because each algorithm is compiled with different parameter settings, such as the termination criterions, the starting points, or the continuation techniques. Hence, in our performance comparisons, we run each code from the same initial point, use all the default parameter values, and only observe the convergence behavior of each algorithm to attain a solution of similar accuracy. We provide below a brief review of each solver.

NESTA[1] uses Nesterov's smoothing technique [28] and gradient method [29] to solve basis pursuit denoising problem. The current version can solve ℓ_1-norm regularization problems with different types, including (1.2). In this experiment, we test NESTA with continuation (hereafter referred to as NESTA-Ct) for comparison. This algorithm solves a sequence of problems (1.2) using a decreasing sequence of μ values. Additionally, NESTA-Ct initially uses the intermediated solution for the next problem. In running NESTA, all the parameters are taken as default except `TolVar`, which is set to $1e-5$, to obtain solutions of similar quality with other solvers.

Gradient projections for sparse reconstruction (referred to as GPSR_BB [2] [20]) reformulates the original problem (1.2) as a box-constrained quadratic programming problem (1.6) by splitting $x = u - v$. Figueiredo, Nowak, and Wright use a gradient projection method with Barziali-Borwein steplength [2] for its solution, the performance of which is improved using a nonmonotone line search [22]. For the comparison of the proposed algorithm with GPSR-BB, we use the former's continuation variant and set all parameters as default.

[1] Available at http://www.acm.caltech.edu/~nesta.

[2] Available at http://www.lx.it.pt/~mtf/GPSR.

The well-known CGD[3] uses a gradient algorithm to solve (1.5) and obtain the search direction $d_k^i = ze^i$ in $i \in \mathcal{J}$, where $\mathcal{J}$ is a nonempty subset of $\{1, \ldots, n\}$. Moreover, CGD chooses the index subset $\mathcal{J}$ following the Gauss-Southwell rule. The iterative process $x_{k+1} = x_k + \alpha_k d_k$ continues until some termination criteria are met, (i.e., $d_k^i = 0$ with $i \notin \mathcal{J}$), and the stepsize α_k using an Armijo rule. In running CGD, we use the code `CGD` according to its Matlab package and set all the parameters as default except for `init=2` to start the iterative process at $A^\top b$.

TwIST[4] is a two-step IST algorithm that is used to solve a class of linear inverse problems. Specifically, TwIST is designed to solve

$$\min_u \mathcal{J}(u) + \frac{\mu}{2}\|Au - f\|_2^2, \tag{5.4}$$

where A is a linear operator, and $\mathcal{J}(\cdot)$ is a general regularizer, which can be either a ℓ_1-norm or total variation [34]. The iteration framework of TwIST is

$$u_{k+1} = (1-\alpha)u_{k-1} + (\alpha - \delta)u_k + \delta\Phi_\mu(\xi_k),$$

where $\alpha, \delta > 0$ are parameters, $\xi_k = u_k + A^\top(f - Au_k)$, and

$$\Phi_\mu(\xi_k) = \arg\min_u \mathcal{J}(u) + \frac{\mu}{2}\|u - \xi_k\|_2^2. \tag{5.5}$$

We use the default parameters in TwIST and terminate the iteration process when the relative variation of function value falls below 10^{-4}.

FPC[5] is used to solve the general ℓ_1-regularized minimization problem (1.1), where f is a continuously differentiable convex function. At current x_k and any scalar $\tau > 0$ (might be $1/\lambda_k$), the next iteration is produced by the so-called fixed point iteration

$$x_{k+1} = \mathcal{S}(x_k - \tau\nabla f(x_k), \mu\tau).$$

To obtain a good practical performance, FPC is augmented with a continuation approach. Moreover, FPC is further modified using Barzilai–Borwein stepsize (code FPC-BB in Matlab package FPC_v2), resulting in a significantly improved performance. In running FPC-BB, we use all the default parameter values except `xtol` $= 1 \cdot \mathrm{e} - 5$ to stop the algorithm when the relative change between successive points is below `xtol` to derive solutions of similar quality with other solvers. The closely related algorithm FPC_AS [6] searches along

$$d_k = \mathcal{S}(x_k - \tau\nabla f(x_k), \mu\tau) - x_k$$

to estimate the support to the solution using the nonmonotone line search of Zhang and Hager [48]. Then, a smooth "subspace optimization" is solved to recover the magnitudes of the nonzero components of x. In running FPC_AS, we use the latest version in its Matlab package and set all the parameters as default except for `opts.init=2` to start at $A^\top b$.

In this test, A is a partial discrete cosine transform (DCT) coefficient matrix, whose m rows are chosen randomly from the $n \times n$ DCT matrix. This encoding matrix A does not require storage and allows fast matrix-vector multiplications involving A and $A^\top$. Therefore, it can be used to test larger-size problems compared with those tested using Gaussian matrices. In NBBL1, we take $tol_2 = 10^{-4}$, $h = 0.83$, $\lambda_{(\min)} = 10^{-30}$, and $\lambda_{(\max)} = 10^{30}$. In the line

[3] Available at http://www.math.nus.edu.sg/~matys/.

[4] Available at: http://www.lx.it.pt/~bioucas/TwIST/TwIST.htm.

[5] Available at http://www.caam.rice.edu/~optimization/L1/fpc.

[6] Available at http://www.caam.rice.edu/~optimization/L1/FPC_AS/.

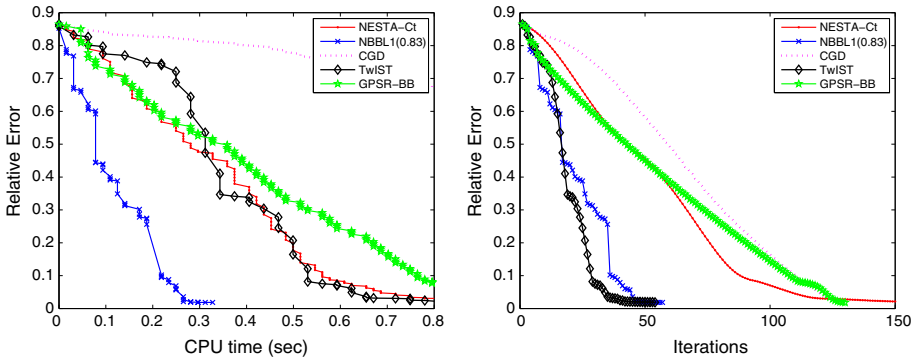

Fig. 3 Comparison result of NBBL1, NESTA_Ct, CGD, TwIST, and GPSR_BB. The x-axes represent the CPU time in seconds (*left*) and the number of iterations (*right*). The y-axes represent the relative error

search, we choose $\tilde{\alpha}_0 = h$, $\rho = 0.35$, $\delta = 10^{-5}$, and $\tilde{m} = 5$. In this comparison, we let $n = 2^{14}$, $m = \texttt{floor}(n/4)$, where "`floor`" is a Matlab command used to round off an element to the nearest integer toward minus infinity. The original signal $\bar{x}$ contains $p = \texttt{floor}(m/8)$ number of nonzero components. Moreover, the observation b is contaminated by Gaussian noise with level $\sigma = 1\text{e}{-3}$. The goal is to use each algorithm to reconstruct $\bar{x}$ from the observation b by solving (1.2) with $\mu = 2^{-8}$. All the tested algorithms start at $x_0 = A^\top b$ and terminate with different stopping criterions to produce resolutions of similar quality. Note that for the continuation technique used in some of the tested algorithms, the value of μ may change occasionally. For this reason, we only observe the convergence behavior of the algorihtms with respect to relative error as the iteration numbers and computing time increase to specifically illustrate the performance of each algorithm.. The results of NBBL1, NESTA_Ct, CGD, TwIST, and GPSR_BB are presented in Fig. 3.

The left plot in Fig. 3 shows that NBBL1 usually decreases the relative errors faster than NESTA_Ct, CGD, GPSR_BB, and TwIST throughout the entire iteration process. Meanwhile, the right plot shows that NBBL1 requires lesser number of iterations than NESTA_Ct, CGD, and GPSR_BB. Notably, TwIST requires a nearly equal number of iterations as NBBL1 but with more computing time. The reason lies in solving the de-noising subproblem (5.5) per-iteration. Among the compared solvers, CGD performs the weakest. However, if CGD starts at zero, its performance should significantly improve (see [43]). In sum, the simple experiment verifies that NBBL1 is the fastest algorithm among those included inthis set of test.

Now we proceed to testing the algorithm against two other related solvers, namely, PFC_BB and FPC_AS, while keeping the experiment settings as the same as previous test. The reason behind our choice to conduct this particular experiment is that the three categories of algorithms all use the Barzilai–Borwein coefficient and nonmonotone line search strategy but in different ways. The search direction of FPC_AS is clearly included in the one defined by (2.8) as a special case. Nevertheless, we still test NBBL1 when $h = 1$ [abbreviated as NBBL1(1.0)], the purpose of which is to demonstrate that the search direction (2.8) would greatly benefit the performance of the algorithm. The results of NBBL1, FPC_BB, and FPC_AS are presented in Fig. 4.

As shown in Fig. 4 all the tested algorithms eventually attain nearly equal relative errors at the end. FPC_BB is the most competitive with NBBL1(0.83), whereas NBBL1(1.0) and FPC_AS are much slower. FPC_AS is the slowest in decreasing relative errors throughout the

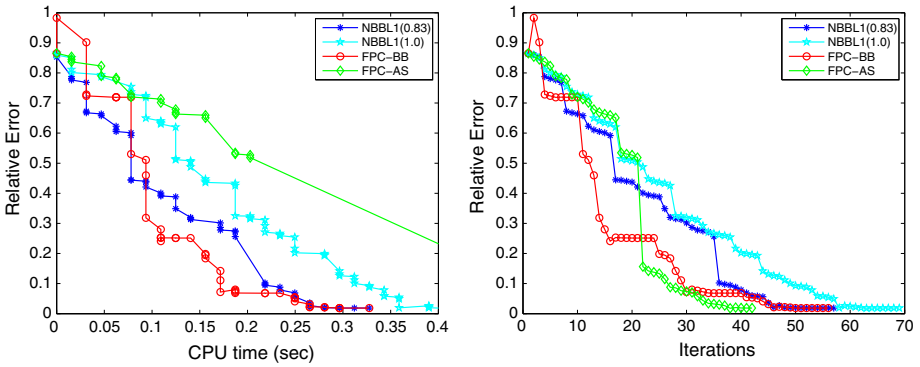

Fig. 4 Comparison result of NBBL1, FPC_AS and FPC_BB. The x-axes represent the CPU time in seconds (*left*) and the number of iterations (*right*). The y-axes represent the relative error

Table 3 Comparison results for each test algorithm

Algorithms	Iter	Time	RelErr	Fun
NBBL1(0.83)	57	0.33	1.8549e−2	1.5708
NBBL1(1.0)	69	0.45	1.8538e−2	1.5708
NESTA_Ct	189	1.23	1.9755e−2	1.5876
GPSR_BB	130	0.89	1.8378e−2	1.6053
TwIST	54	1.05	1.8846e−2	1.5708
CGD	131	2.73	1.8958e−2	1.5708
FPC_BB	56	0.33	1.8539e−2	1.5708
FPC_AS	42	0.58	1.8674e−2	1.5708

entire iteration process, but requires the least number of iterations, because at each iteration, FPC_AS solves a subspace minimization problem. One fact that must be emphasized is that the performance of NBBL1 indeed improves dramatically using $h = 0.83$. In conclusion, the proposed algorithm is generally more efficient than FPC_AS and competes strongly with FPC_BB.

To further illustrate the benefit of NBBL1, we present the comparison results of each algorithm behind the previous two experiments in Table 3 at the end of this section. In this comparison, we only consider the iteration numbers (Iter), the computing time (Time), and the relative error (RelErr) and the function values (Fun) at the final solutions for each algorithm.

It can be seen from Table 3 that, each algorithm obtained solutions with comparable objective function values and comparable relative errors. From the results, once again we see that FPC_BB is most competitive with NBBL1(0.83), while others are relatively slow. Based on the experimental results, NBBL1(1.0) is appreciably slower than NBBL1(0.83). We also did a series of tests with different of experiments settings and observed consistent results. Specifically, FPC_BB is the most competitive with NBBL1(0.83). These results and observations sufficiently demonstrated the efficiency and stability of NBBL1.

6 Conclusions

In this study, we proposed, analyzed, and tested a new practical algorithm to solve the separable nonsmooth minimization problem consisting of an ℓ_1-norm regularized term and

a continuously differentiable term. This type of problem mainly appears in signal/image processing, compressive sensing, machine learning, and linear inverse problems. However, the problem is challenging because of the non-smoothness of the regularization term. Our approach minimizes an approximal local quadratic model to determine a search direction at each iteration. We showed that the search direction contains the one of FPC_AS as a special case, and is reduced to the classic Barzilai–Borwein gradient method when $\mu = 0$. We also proved that the objective function is descent along the direction provided that the initial stepsize does not exceed h in the nonmonotone line search step. We established the algorithm's global convergence theorem by assuming that f is bounded below. Extensive experimental results illustrated that the proposed algorithm is an effective tool to solve ℓ_1-regularized nonconvex problems from CUTEr library. Moreover, we ran our algorithm to recover a large sparse signal from its noisy measurement. The performance comparisons with several state-of-the-art solvers verified that our algorithm is competitive with FPC_BB and faster than FPC_AS, NESTA_Ct, CGD, GPSR, and TwIST.

Unlike all the existing algorithms in the literature, our approach uses a linear model to approximate $\|x_k + d\|_1$ for computing the search direction with a small scalar h; that is,

$$\|x_k + d\|_1 \approx \|x_k\|_1 + \frac{\|x_k + hd\|_1 - \|x_k\|_1}{h}.$$

Although the equations may hold exactly when $h = 1$, a series of numerical experiments in this paper showed that an appropriate h may result in an improved performance with suitable experiment settings. This approach is distinctive and novel; therefore, it is one of the important contributions of this study. A natural question is whether the performance of FPC_AS, even its related variants, can be improved using the search direction defined in (2.8). The answer deserves a thorough investigation. The nonmonotone Barzilai–Borwein gradient algorithm of Raydan [32] is known to be very effective in smooth unconstrained minimization, and its equally remarkable effectiveness in signal reconstruction problems involving ℓ_1-regularized problems has not been clearly explored. Hence, our approach can be considered as a modification or extension of the algorithm in [32]. Moreover, the numerical experiments illustrated that our approach is comparable with or even better than several state-of-the-art algorithms. This advantage certainly serves as the numerical contribution of this study. Our algorithm readily solves the ℓ_1-regularized logistic regression, the ℓ_2-norm, as well as matrix trace norm and $\ell_{2,1}$-norm minimization problems in machine learning. However, tests related to these problems were not conducted in the study. Further investigations are therefore necessary.

Acknowledgments We would like to thank two anonymous referees for their useful comments and suggestions which improved this paper greatly. The first version of the paper is finished during Y. Xiao's stay as a postdoctoral research fellow in NCTS, National Cheng Kung University, Taiwan. The work of Y. Xiao is supported by Chinese Natural Science Foundation Grant 11001075, and the Natural Science Foundation of Henan Province Grant 2011GGJS030.

References

1. Andrew, G., Gao, J.: Scalable training of ℓ_1-regularized log-linear models. In: Proceedings of the Twenty Fourth International Conference on Machine Learning, (ICML) (2007)
2. Barzilai, J., Borwein, J.M.: Two point step size gradient method. IMA J. Numer. Anal. **8**, 141–148 (1988)
3. Beck, A., Teboulle, M.: A fast iterative shrinkage-thresholding algorithm for linear inverse problems. SIAM J. Imaging Sci. **2**, 183–202 (2009)

4. Becker, S., Bobin, J., Candès, E.: NESTA: a fast and accurate first-order method for sparse recovery. SIAM J. Imaging Sci. **4**, 1–39 (2011)
5. Birgin, E.G., Martínez, J.M., Raydan, M.: Nonmonotone spectral projected gradient methods on convex sets. SIAM J. Optim. **10**, 1196–1211 (2000)
6. Bioucas-Dias, J.M., Figueiredo, M.: A new TwIST: two-step iterative shrinkage/thresholding algorithms for image restoratin. IEEE Trans. Image Process. **16**, 2992–3004 (2007)
7. Cai, J.F., Candès, E., Shen, Z.: A singular value thresholding algorithm for matrix completion. SIAM J. Optim. **20**, 1956–1982 (2010)
8. Candès, E., Romberg, J.: Quantitative robust uncertainty principles and optimally sparse decompositions. Found. Comput. Math. **6**, 227–254 (2006)
9. Candès, E., Romberg, J., Tao, T.: Stable signal recovery from imcomplete and inaccurate information. Commun. Pure Appl. Math. **59**, 1207–1233 (2005)
10. Candès, E., Romberg, J., Tao, T.: Robust uncertainty principles: exact signal reconstruction from highly incomplete frequence information. IEEE Trans. Inf. Theory **52**, 489–509 (2006)
11. Candès, E., Tao, T.: Near optimal signal recovery from random projections: universal encoding strategies. IEEE Trans. Inf. Theory **52**, 5406–5425 (2004)
12. Chang, K.W., Hsieh, C.J., Lin, C.J.: Coordinate descent method for large-scale L2-loss linear SVM. J. Mach. Learn. Res. **9**, 1369–1398 (2008)
13. Cheng, W., Li, D.H.: A derivative-free nonmonotone line search and its application to the spectral residual method. IMA J. Numer. Anal. **29**, 814–825 (2009)
14. Conn, A.R., Gould, N.I.M., Toint, PhL: CUTE: constrained and unconstrained testing environment. ACM Trans. Math. Softw. **21**, 123–160 (1995)
15. Dai, Y.H., Fletcher, R.: On the asymptotic behaviour of some new gradient methods. Math. Program. **103**, 541–559 (2005)
16. Dai, Y.H., Hager, W.W., Schittkowski, K., Zhang, H.C.: The cyclic Barzilai–Borwein method for unconstrained optimization. IMA J. Numer. Anal. **26**, 604–627 (2006)
17. Dai, Y.H., Liao, L.Z.: R-linear convergence of the Barzilai and Borwein gradient method. IMA J. Numer. Anal. **26**, 1–10 (2002)
18. Donoho, D.L.: Compressed sensing. IEEE Trans. Inf. Theory **52**, 1289–1306 (2006)
19. Duchi, J., Singer, Y.: Efficient online and batch learning using forward backword splitting. J. Mach. Learn. Res. **10**, 2899–2934 (2009)
20. Figueiredo, M., Nowak, R.D., Wright, S.J.: Gradient projection for sparse reconstruction: application to compressed sensing and other inverse problems. IEEE J. Sel. Top. Signal Process **1**, 586–597 (2007)
21. Genkin, A., Lewis, D.D., Madigan, D.: Large-scale Bayesian logistic regression for text categorization. Technometrices **49**, 291–304 (2007)
22. Grippo, L., Lampariello, F., Lucidi, S.: A nonmonotone line search technique for Newton's method. SIAM J. Numer. Anal. **23**, 707–716 (1986)
23. Hale, E.T., Yin, W., Zhang, Y.: Fixed-point continuation for ℓ_1-minimization: methodology and convergence. SIAM J. Optim. **19**, 1107–1130 (2008)
24. Kim, S., Koh, K., Lustig, M., Boyd, S., Gorinevsky, D.: An interior-point method for large-scale ℓ_1-regularized least squares. IEEE J. Sel. Top. Signal Process. **1**, 606–617 (2007)
25. Koh, K., Kim, S., Boyd, S.: An interior-point method for large-scale ℓ_1-regularized logistic regression. J. Mach. Learn. Res. **8**, 1519–1555 (2007)
26. Lin, C.J., Moré, J.J.: Newton's method for large-scale bound constrained problems. SIAM J. Optim. **9**, 1100–1127 (1999)
27. Lu, Z., Zhang, Y.: An augmented Lagrangian approach for sparse principal component analysis. Math. Program. **135**, 149–193 (2012)
28. Nesterov, Y.: Smooth minimization of non-smooth functions. Math. Program. **103**, 127–152 (2005)
29. Nesterov, Y.: Gradient methods for minimizing composite objective function, ECORE Discussion Paper 2007/76. http://www.ecore.be/DPs/dp_1191313936.pdf (2007)
30. Nocedal, J.: Updating quasi-Newton matrices with limited storage. Math. Comput. **35**, 773–782 (1980)
31. Raydan, M.: On the Barzilai and Borwein choice of steplength for the gradient method. IMA J. Numer. Anal. **13**, 321–326 (1993)
32. Raydan, M.: The Barzilai and Borwein gradient method for the large scale unconstrained minimization problem. SIAM J. Optim. **7**, 26–33 (1997)
33. Recht, B., Fazel, M., Parrilo, P.A.: Guaranteed minimum rank solutions of matrix equations via nuclear norm minimization. SIAM Rev. **52**, 471–501 (2010)
34. Rudin, L., Osher, S., Fatemi, E.: Nonlinear total variation based noise removal algorithms. Phys. D Nonlinear Phenom. **60**, 259–268 (1992)

35. Shalev-Shwartz, S., Tewari, A.: Stochastic method for l1 regularized loss minimization. In: Proceedings of the Twenty Sixth International Conference on Machine Learning (ICML) (2009)
36. Shi, J., Yin, W., Osher, S., Sajda, P.: A fast hybrid algorithm for large-scale ℓ_1-regularized logistic regression. J. Mach. Learn. Res. **11**, 713–741 (2010)
37. Tseng, P., Yun, S.: A coordinate gradient descent method for nonsmooth separable minimization. Math. Program. **117**, 387–423 (2009)
38. van den Berg, E., Friedlander, M.P.: Probing the Pareto frontier for basis pursuit solutions. SIAM J. Sci. Comput. **31**, 890–912 (2008)
39. Wen, Z., Yin, W., Goldfarb, D., Zhang, Y.: A fast algorithm for sparse reconstruction based on shrinkage, subspace optimization, and continuation. SIAM J. Sci. Comput. **32**, 1832–1857 (2010)
40. Wen, Z., Yin, W., Zhang, H., Goldfarb, D.: On the convergence of an active-set method for ℓ_1 minimization. Optim. Method Softw **27**, 1127–1146 (2012)
41. Wright, S.J., Nowak, R.D., Figueiredo, M.A.T.: Sparse reconstruction by separable approximation. In: Proceedings of the International Conference on Acoustics, Speech, and Signal Processing, pp 3373–3376 (2008)
42. Wright, J., Ma, Y., Ganesh, A., Rao, S.: Robust principal component analysis: exact recovery of corrupted low-rank matrices via convex optimization. J. ACM **5**, 1–44 (2009)
43. Yang, J., Zhang, Y.: Alternating direction algorithms for ℓ_1-problems in compressive sensing. SIAM J. Sci. Comput. **33**, 250–278 (2011)
44. Yu, J., Vishwanathan, S.V.N., Günter, S., Schraudolph, N.N.: A quasi-Newton approach to nonsmooth convex optimization problems in machine learning. J. Mach. Learn. Res. **11**, 1145–1200 (2010)
45. Yuan, G.X., Chang, K.W., Hsieh, C.J., Lin, C.J.: A comparison of optimization methods and software for large-scale ℓ_1-regularized linear classification. J. Mach. Learn. Res. **11**, 3183–3234 (2010)
46. Yuan, X.: Alternating direction method for covariance selection models. J. Sci. Comput. **51**, 261–273 (2012)
47. Yun, S., Toh, K.C.: A coordinate gradient descent method for ℓ_1-regularized convex minimization. Comput. Optim. Appl. **48**, 273–307 (2011)
48. Zhang, H., Hager, W.W.: A nonmonotone line search technique and its application to unconstrained optimization. SIAM J. Optim. **14**, 1043–1056 (2004)
49. Zhang, Y., Sun, W., Qi, L.: A nonmonotone filter Barzilai-Borwein method for optimization. Asia Pac. J. Oper. Res. **27**, 55–69 (2010)