Calcolo (2019) 56:1
https://doi.org/10.1007/s10092-018-0296-x

A new generalized shrinkage conjugate gradient method for sparse recovery

Hamid Esmaeili[1] · **Shima Shabani**[1] · **Morteza Kimiaei**[2]

Received: 20 February 2018 / Accepted: 12 November 2018 / Published online: 3 December 2018

Abstract

In this paper, a new procedure, called *generalized shrinkage conjugate gradient* (GSCG), is presented to solve the ℓ_1-regularized convex minimization problem. In GSCG, we present a new descent condition. If such a condition holds, an efficient descent direction is presented by an attractive combination of a generalized form of the conjugate gradient direction and the ISTA descent direction. Otherwise, ISTA is improved by a new step-size of the shrinkage operator. The global convergence of GSCG is established under some assumptions and its sublinear (R-linear) convergence rate in the convex (strongly convex) case. In numerical results, the suitability of GSCG is evaluated for compressed sensing and image debluring problems on the set of randomly generated test problems with dimensions $n \in \{2^{10}, \ldots, 2^{17}\}$ and some images, respectively, in Matlab. These numerical results show that GSCG is efficient and robust for these problems in terms of the speed and ability of the sparse reconstruction in comparison with several state-of-the-art algorithms.

Keywords ℓ_1-Minimization · Compressed sensing · Image debluring · Shrinkage operator · Generalized conjugate gradient method · Nonmonotone technique · Line search method · Global convergence

Mathematics Subject Classification 65K05 · 90C25 · 90C06 · 94A08

✉ Hamid Esmaeili
esmaeili@basu.ac.ir

Shima Shabani
sh.shabani@basu.ac.ir

Morteza Kimiaei
kimiaeim83@univie.ac.at
http://www.mat.univie.ac.at/~kimiaei/

[1] Department of Mathematics, Bu-Ali Sina University, Hamedan, Iran

[2] Faculty of Mathematics, University of Vienna, Oskar-Morgenstern-Platz 1, A-1090 Vienna, Austria

1 Introduction

Consider the following unconstrained optimization problem for the sparse recovery

$$\begin{array}{ll} \min & F(x) := f(x) + \mu\|x\|_1, \\ \text{s.t.} & x \in \mathbb{R}^n \end{array} \tag{1}$$

where $f : \mathbb{R}^n \to \mathbb{R}$ is a smooth convex function, $\|\cdot\|_1$ is the ℓ_1-norm of a vector $x \in \mathbb{R}^n$, usually called *regularizer* or *regularization function*, and $\mu \in \mathbb{R}^+$ is a regularization parameter that can be interpreted as a trade-off parameter or relative weight between the objective terms. The problem (1) generalizes the well known *basis pursuit denoising* (BPDN) or $\ell_2 - \ell_1$ problem, found in the signal and image processing literature, as

$$\begin{array}{ll} \min & F(x) := \frac{1}{2}\|Ax - b\|^2 + \mu\|x\|_1. \\ \text{s.t.} & x \in \mathbb{R}^n \end{array} \tag{2}$$

In (2), $\|\cdot\|$ stands for the standard *Euclidean* norm, $A \in \mathbb{R}^{m\times n}$ ($m \ll n$) and $b \in \mathbb{R}^m$. As a basic idea, a sparse solution of the underdetermined linear system $Ax = b$ may be obtained by the ℓ_0-norm optimization problem. This nonconvex combinatorial problem is NP-hard [23] and hence is difficult to solve. Instead of solving it, Candès et al. [12] and Donoho [15] suggested the ℓ_1-norm convex relaxation problem to recover the sparse solution under some conditions. However, the problem (2) is one of the most famous models in sparse recovery area which gives the sparsest solution of the aforementioned underdetermined linear system. This model is a robust version of reconstruction process when the measurements are contaminated with noise. Some popular applications for the problem (2) are the areas of *compressed sensing* (CS) and *image debluring* (ID) problems. For more information about these applications, readers can refer to [17,18,23].

Contribution This paper aims to present an innovative algorithm based on a new direction which is descendant under a descent condition and a new shrinkage step-size in order to accelerate `ISTA`. Such an algorithm generates a new generalized form of the CG direction and uses it under a mild norm condition. Moreover, it uses the `ISTA` descent direction with the improved shrinkage step-size, produced based on a pseudo CG idea, whenever the generated direction is not descendant or does not satisfy the norm condition.

Organization The remaining of this article is organized as follows. In Sect. 2, advantages and shortcomings of some algorithms and software are investigated. We have a review of `ISTA`, a gradient based method, in Sect. 3. Section 4 is dedicated to our method, which has three subsections: presenting the new `GSCG` approach in detail in the first subsection, presenting the acceptance criterion of the new method based on a modified nonmonotone Armijo line search strategy in the second subsection and finally presenting a flowchart and a pseudo code of the new algorithm in the third subsection. The global convergence and sublinear/R-linear convergence rate of the proposed algorithm are analyzed in Sect. 5. In Sect. 6, numerical results on CS and ID problems are reported. At the end, in Sect. 7, some conclusions are presented.

Preliminaries We first recall some definitions, which will play a key role in the convex optimization analysis.

Definition 1 A *directional derivative* of a function $F : \mathbb{R}^n \to \mathbb{R}$ at point $x \in \mathbb{R}^n$ along the direction $d \in \mathbb{R}^n$, denoted by $F'(x; d)$, is given by the following limit if it exists:

$$F'(x; d) := \lim_{\lambda \downarrow 0} \frac{F(x + \lambda d) - F(x)}{\lambda}.$$

The directional derivative may be well defined even when F is not continuously differentiable. In fact, it is the most useful one in such a situation.

Definition 2 A point $x^* \in \mathbb{R}^n$ is called a *stationary point* of a function $F : \mathbb{R}^n \to \mathbb{R}$, if $F'(x^*; d) \geq 0$ for all $d \in \mathbb{R}^n$, see [38, page 394].

Definition 3 Suppose that $F : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ is a closed proper convex function and ς is a positive parameter. A *proximal operator* $\mathbf{Prox}_{\varsigma F} : \mathbb{R}^n \to \mathbb{R}^n$ of the scaled function ςF at $y \in \mathbb{R}^n$ is defined by

$$\mathbf{Prox}_{\varsigma F}(y) := \operatorname{argmin}_x \left\{ F(x) + \frac{1}{2\varsigma} \|x - y\|^2 \right\},$$

where parameter ς controls the relative weight of the two terms. For more information about the proximal operator and its algorithms, we refer to [35] which is a monograph about this class of optimization algorithms.

Notation Throughout this paper, for the convenience of notation, write $F_k := F(x_k)$ and $f_k := f(x_k)$ and let $\nabla f_k := \nabla f(x_k)$ be the *gradient* of f at x_k. The subscript k often represents the iteration number in an algorithm and D_S denotes the diameter of a bounded set S.

2 Related algorithms

There are a lot of solvers for nonsmooth optimization. Here, we review only some of them, related to our study as tested in numerical results section. The objective function of problem (1) is convex but nonsmooth since it includes the ℓ_1-norm term. This problem can be transformed to a linear optimization problem and solved via standard tools such as simplex or interior point methods [7,34]. These methods have a high computational cost because of using a dense data structure, especially for large-scale problems. For large problems, there are gradient-based algorithms, as well, which calculate the direction approximately without computing or approximating the Hessian, using gradients only. Despite requiring little storage, they have so slow convergence properties; see [11,13,16,19,21,27,28,40]. One popular method of this class is *iterative shrinkage-thresholding algorithm* (`ISTA`) [13,16] with the general step

$$x_{k+1} = \mathcal{S}_{\mu\tau_k}\Big(x_k - \tau_k \nabla f_k\Big), \tag{3}$$

where τ_k is the *shrinkage step-size* and $\mathcal{S}_\nu : \mathbb{R}^n \to \mathbb{R}^n$ is the *shrinkage operator*, defined by

$$\mathcal{S}_\nu(x) := \text{sgn}(x) \odot \max\left\{|x| - \nu, 0\right\},$$

in which $\nu > 0$, $\text{sgn}(\cdot)$ stands for the *signum function* and $\odot$ denotes the *component-wise product*, i.e., $(x \odot y)_i := x_i y_i$. The important characteristic of `ISTA` is its simple form. However, `ISTA` has been characterized as a slow method and its convergence analysis has been well studied in [8,13,16,20], showing its sublinear convergence rate. Some methods have been proposed in order to accelerate `ISTA` in the past years. We consider some of them as follows:

(i) Fixed-point continuation algorithm (`fpc`) [27] uses an operator-splitting technique to solve a sequence of problem (1), defined by a decreasing sequence of parameter $\{\mu_j\}$ with the fixed shrinkage step-size. In `fpc`, based on the continuation (homotopy) strategy, the solution of the current problem is used as the initial estimate of the solution to the next problem.
(ii) `fpcbb` [28] is a fixed-point continuation algorithm for solving (1) which constructs its shrinkage step-size by Barzilai–Borwein (BB) technique [6].
(iii) `NBBL1` [42] is a nonmonotone Barzilai–Borwein gradient algorithm for solving (1). At each step, such an algorithm generates the search direction by minimizing a local approximal quadratic model of (1) which contains an approximated form of the ℓ_1-regularization term, due to its non-differentiability.
(iv) `TwIST` [11] is a two-step `ISTA` which relies on computing the next iteration based on two previously computed steps instead of one previous step.
(v) `FISTA` [8] is a fast iterative shrinkage-thresholding algorithm, employed on a special linear combination of the previous two iterations and this is its main difference from `ISTA`. This method keeps computational simplicity of `ISTA` and improves its global convergence rate to $\mathcal{O}(\frac{1}{k^2})$, comparable to `ISTA` with $\mathcal{O}(\frac{1}{k})$.
(vi) `SpaRSA` [41] is a sparse reconstruction by separable approximation which solves (1) with separable structures. Improved practical performance of `SpaRSA` results in the variation of the shrinkage step-size.

3 Review of `ISTA`

The classical steepest descent method is the simplest approach to solve (1), in the case $\mu = 0$. This method produces a sequence $\{x_k\}_{k\geq 0}$ via

$$x_{k+1} := x_k - \tau_k \nabla f_k, \tag{4}$$

for which $\tau_k > 0$ is a suitable step-size. The gradient iteration (4) can be converted into the following iterative scheme, known as the *proximal regularization* of the linearized function f at x_k,

$$x_{k+1} := \arg\min_x \left\{ f_k + \langle x - x_k, \nabla f_k \rangle + \frac{1}{2\tau_k}\|x - x_k\|^2 \right\}. \tag{5}$$

By applying the structure of (5) for (1) and setting $d := x - x_k$, we get

$$x_{k+1} = \arg\min_d \left\{ f_k + \langle d, \nabla f_k \rangle + \frac{1}{2\tau_k}\|d\|^2 + \mu\|x_k + d\|_1 \right\}. \tag{6}$$

Having ignored the constant terms, we can rewrite (6) as

$$x_{k+1} = x_k + d_k^s = \arg\min_d \left\{ \frac{1}{2}\|x_k + d - (x_k - \tau_k \nabla f_k)\|^2 + \mu\tau_k\|x_k + d\|_1 \right\}. \tag{7}$$

As shown in [35], the favorable structure of (7) adopts the explicit solution (3) which leads to the ISTA iteration for the problem (1). Both (3) and (7) yield to

$$d_k^s := \mathcal{S}_{\mu\tau_k}\Big(x_k - \tau_k \nabla f_k\Big) - x_k, \tag{8}$$

called *shrinkage direction* or ISTA *direction*. The following lemma shows that the shrinkage direction, obtained by (8), is a descent direction whenever $d_k^s \neq 0$.

Lemma 1 *Suppose that $\tau_k > 0$ and d_k^s is the shrinkage direction (8). Then, we have*

$$F\left(x_k + \alpha d_k^s\right) \leq F_k + \alpha\Big[\left\langle \nabla f_k, d_k^s \right\rangle + \mu\Big(\left\|x_k + d_k^s\right\|_1 - \left\|x_k\right\|_1\Big)\Big] + o(\alpha),$$

for all $\alpha \in (0, 1]$ and

$$\Delta_k^s := \left\langle \nabla f_k, d_k^s \right\rangle + \mu\Big(\left\|x_k + d_k^s\right\|_1 - \left\|x_k\right\|_1\Big) \leq -\frac{1}{2\tau_k}\|d_k^s\|^2. \tag{9}$$

Proof The proof can be found in [38]. □

The next lemma, the proof of which can be found in [38], characterizes the situation of $d_k^s = 0$.

Lemma 2 *Suppose that $\tau_k > 0$ and d_k^s is the shrinkage direction (8). Then, x_k is a stationary point for the problem (1) if and only if $d_k^s = 0$.*

It is worth noting that ISTA uses a more conservative choice of the shrinkage step-size τ_k, related to the Lipschitz constant of ∇f. Furthermore, in some fixed point methods like [27], τ_k is considered to be fixed so that these methods could not produce a suitable step-size close to the optimizer or far away from it. Hence, a bad value of τ_k leads to a slow convergence rate. To overcome this disadvantage, in [28,40–42], τ_k is chosen dynamically by the BB method. BB method is an accelerated form of the classical steepest descent method using two consecutive iterations to introduce the step-size of the next step, i.e., $x_{k+1} := x_k - \tau_k^{\text{bb}} \nabla f_k$, where τ_k^{bb}, called *BB step-size*, is presented as one of the following

$$\tau_k^{\text{bb},1} := \frac{\langle s_{k-1}, s_{k-1}\rangle}{\langle s_{k-1}, y_{k-1}\rangle} \quad \text{or} \quad \tau_k^{\text{bb},2} := \frac{\langle s_{k-1}, y_{k-1}\rangle}{\langle y_{k-1}, y_{k-1}\rangle}, \tag{10}$$

where $s_{k-1} := x_k - x_{k-1}$, $y_{k-1} := \nabla f_k - \nabla f_{k-1}$. In (11), $0 < \tau_{\min} < \tau_{\max} < \infty$, known as the *safeguard parameters*, are used to prevent the production of a very small or large BB step-size. Such an improved BB step-size is presented by

$$\tau_k^{\text{sbb},i} := \max\left\{\tau_{\min}, \min\left\{\tau_k^{\text{bb},i}, \tau_{\max}\right\}\right\}, \tag{11}$$

where $i = 1, 2$. To simplify our notation, we set $\tau_k^{\text{sbb}} := \tau_k^{\text{sbb},i}$.

4 Our method

In this section, an algorithmic framework of the new approach is presented to solve large-scale nonsmooth convex optimization problems. This section is followed in three subsections. In the first subsection, the new method is introduced in details. In the second subsection, determining the step-size $\alpha_k \in (0, 1]$, we utilize a modified nonmonotone Armijo line search strategy for nonsmooth convex optimization problems and in the third subsection, the flowchart and the detailed pseudo code of the new method are presented.

4.1 New generalized shrinkage conjugate gradient approach (GSCG)

Accelerating the proximal gradient methods such as ISTA, one can use the idea of *momentum term* $\Theta_k(x_k - x_{k-1})$, through which the next step x_{k+1} depends on the two previous steps x_k and x_{k-1} with $\Theta_k > 0$. FISTA and TwIST, accelerated forms of ISTA, use this idea in order to solve (1), see [8,11]. It is remarkable that CG methods [22,26,29,36], the improved forms of the descent gradient method, have a momentum term in their common iterative schemes. In addition, in [33], a generalized form of the nonlinear CG methods has been presented where the negative gradient is replaced with a general descent direction. Our goal here is to accelerate ISTA by either strengthening its descent direction or improving the shrinkage step-size. In some iterations of the GSCG iterative scheme, a pseudo CG direction is produced where the negative gradient-based term of the CG direction is replaced with the descent direction d_k^s in (8) as follows,

$$x_{k+1} := x_k + \alpha_k d_k, \tag{12}$$

$$d_k := d_k^s + \beta_k d_{k-1}, \tag{13}$$

where α_k is a suitable step-size and $\{\beta_k\}_{k\geq 0}$, named CG parameter, is a slowly diminishing constant sequence with $\lim_{k\to\infty} \beta_k = 0$. This CG parameter is defined by

$$\beta_k := \frac{1}{(k+1)^{\lambda}}, \tag{14}$$

where $\lambda \in (0, 1)$. Relation (14) is different from the conventional CG parameters and is similar to the β_k presented in [31,32] for a convex optimization problem over a fixed-point set of a nonexpansive mapping.

The following remark shows the existence of a momentum term in some iterations of GSCG.

Remark 1 Substituting k with $k-1$ in (12) leads to

$$d_{k-1} = \frac{x_k - x_{k-1}}{\alpha_{k-1}}. \tag{15}$$

Then, by replacing (13) and (15) in (12), we obtain

$$x_{k+1} = x_k + \alpha_k d_k^s + \underbrace{\frac{\alpha_k \beta_k}{\alpha_{k-1}}(x_k - x_{k-1})}_{:= \text{momentum term}},$$

Since (13) is not descendant, in general, we impose a *mild descent condition* on this direction, leading to its descent property. This condition is defined by

$$\texttt{dCon} := (\langle \nabla f_k, d_{k-1}\rangle + \mu \|d_{k-1}\|_1 < 0). \tag{16}$$

The following lemma shows that the direction defined by (13) is descendant whenever $d_k \neq 0$.

Lemma 3 *Suppose that $\tau_k > 0$, $\beta_k \geq 0$ and d_k is generated by (13). If the condition (16) holds, then we have*

$$F(x_k + \alpha d_k) \leq F_k + \alpha \Delta_k^{\text{cg}} + o(\alpha), \quad \forall \alpha \in (0, 1], \tag{17}$$

and

$$\Delta_k^{\text{cg}} := \langle \nabla f_k, d_k\rangle + \mu\Big(\|x_k + d_k\|_1 - \|x_k\|_1\Big) \leq 0. \tag{18}$$

Proof The proof of (17) is similar to the first part of Lemma 1, presented in [38]. To prove (18), we consider three cases:

(i) If $\beta_k = 0$, then (18) is satisfied based on Lemma 1.

(ii) If $\beta_k > 0$ and $d_k \neq 0$, then from the definition of d_k, Lemma 1 and (16), we get

$$\begin{aligned}\Delta_k^{\text{cg}} &= \langle \nabla f_k, d_k\rangle + \mu\Big(\|x_k + d_k\|_1 - \|x_k\|_1\Big)\\ &\leq \langle \nabla f_k, d_k^s\rangle + \mu\Big(\|x_k + d_k^s\|_1 - \|x_k\|_1\Big) + \beta_k(\langle \nabla f_k, d_{k-1}\rangle\\ &\quad + \mu\|d_{k-1}\|_1) < 0,\end{aligned}$$

which shows that $d_k \neq 0$ is a descent direction for $F(x)$ at x_k.

(iii) If $\beta_k > 0$ and $d_k = 0$, then the proof is trivial. □

Whenever the descent condition (16) is not satisfied, GSCG takes advantages of the descent direction d_k^s with a new form of τ_k, called *CG step-size* and introduced by

$$\tau_k := \tau_k^{\text{sbb}} + \omega\gamma_k\tau_{k-1}, \tag{19}$$

In (19), τ_k^{sbb} is computed using (11), $\omega \in (0, 1)$ is an impact factor for controlling the effect of $\gamma_k\tau_{k-1}$ and diminishing constant sequence $\{\gamma_k\}_{k\geq 0}$ is defined by

$$\gamma_k := \frac{1}{(k+1)^{\upsilon}}, \tag{20}$$

where $\upsilon \in (0, 1)$. Note that τ_k, generated by (19), is a special linear combination of τ_k^{sbb} and τ_{k-1}, and its structure is similar to that of the CG direction (13). It is clear that $\tau_k^{\text{sbb}} + \omega\gamma_k\tau_{k-1} \geq \tau_k^{\text{sbb}}$. In addition, $\|d_k^s\|$ is nondecreasing in τ_k for each x_k; see [40]. Thus, Lemma 1 guarantees the descent property of d_k^s, produced by (19).

In GSCG, in order to investigate the convergence rate, we impose a mild *norm condition* on using (13) whenever (16) is satisfied as follows,

$$\texttt{nCon} := (\|d_k^s\| \geq \|d_k\|). \tag{21}$$

When (21) is not satisfied, the descent direction d_k^s is utilized, using (19). Let us define the event

$$\texttt{cGrad} := (\texttt{dCon} \ \& \ \texttt{nCon}).$$

Iteration k is said to be a *CG iteration* with the BB step-size (11) if the event cGrad occurs; i.e., cGrad := 1. Otherwise, the iteration will be ISTA with the CG step-size (19) (cGrad := 0). In other words, cGrad, dCon and nCon are the integral parameters. According to the mentioned information, for the iterative scheme (12), we now represent

$$d_k := \begin{cases} d_k^s + \beta_k d_{k-1} & \text{if cGrad,} \\ d_k^s & \text{otherwise.} \end{cases} \tag{22}$$

with

$$\tau_k := \begin{cases} \tau_k^{\text{sbb}} & \text{if cGrad,} \\ \tau_k^{\text{sbb}} + \omega\gamma_k\tau_{k-1} & \text{otherwise.} \end{cases} \tag{23}$$

Based on Lemmas 1–3, whether the event cGrad happens or not, we have the following inequality for $d_k \neq 0$ in (22) which confirms its descent property,

$$\Delta_k \leq -\frac{1}{2\tau_k}\|d_k\|^2 < 0, \tag{24}$$

where Δ_k is defined by

$$\Delta_k := \begin{cases} \Delta_k^{\text{cg}} & \text{if cGrad,} \\ \Delta_k^s & \text{otherwise.} \end{cases} \tag{25}$$

4.2 Acceptance criterion

Monotone methods (descent methods) generate a sequence of iterations such that the corresponding sequence of function values is monotonically decreasing in (1). Their strategies may trap the iterations to the bottom of a curved narrow valley of the objective function which makes the iterative algorithms lose their efficiency.

In contrast, in nonmonotone line search techniques, some growth in function value is permitted which overcomes the difficulty mentioned above and improves convergence speed, see [3–5,24,43]. Let us define $l(k)$ as an integer satisfying $k - m(k) \leq l(k) \leq k$, in which $m(0) = 0$ and $0 \leq m(k) \leq \min\{m(k-1)+1, N-1\}$, with $N > 0$. To increase the efficiency of the new algorithm and to guarantee its global convergence, we use a modified form of the nonmonotone Armijo scheme suggested in [4] as follows,

$$F(x_k + \alpha_k d_k) \leq R_k + \sigma \alpha_k \Delta_k, \tag{26}$$

where

$$R_k := \eta_k F_{l(k)} + (1 - \eta_k) F_k, \tag{27}$$

$\eta_k \in [\eta_{\min}, \eta_{\max}]$, $\eta_{\min} \in [0, 1]$, $\eta_{\max} \in [\eta_{\min}, 1]$, and $\sigma \in (0, 1)$ is a constant, usually chosen to be close to zero. The step-size α_k is the largest member of $\{s, \varrho s, \ldots\}$, with $s > 0$, $\varrho \in (0, 1)$, and

$$F_{l(k)} := \max_{0 \leq j \leq m(k)} \{F_{k-j}\}. \tag{28}$$

This nonmonotone line search method, modified for the nonsmooth convex optimization problem (1), uses a stronger nonmonotone technique whenever iterations are far away from the optimizer and a weaker nonmonotone strategy whenever iterations are close to the optimizer by using an adaptive value for η_k, which can improve convergence results.

The procedure of modified nonmonotone Armijo strategy (NMA) is described below.

Procedure Nonmonotone Armijo(NMA)

Input: x_k, d_k, σ, ϱ, Δ_k, η_k, $m(k)$, s and N.

begin
  $\alpha_k := s$;
  compute $F(x_k + \alpha_k d_k)$, $F_{l(k)}$ by (28) and R_k by (27);
  while $F(x_k + \alpha_k d_k) > R_k + \sigma\, \alpha_k \Delta_k$ **do**
    $\alpha_k := \varrho \alpha_k$;
    compute $F(x_k + \alpha_k d_k)$;
  end
  $x_{k+1} := x_k + \alpha_k d_k$, $F_{k+1} := F(x_{k+1})$, $\nabla f_{k+1} := \nabla f(x_{k+1})$;
  update η_{k+1} using an adaptive formula;
  choose $m(k+1) \in [0, \min\{m(k)+1, N\}]$;
end

Output: x_{k+1}, F_{k+1}, ∇f_{k+1}, η_{k+1} and $m(k+1)$.

In NMA, the *while loop* is usually called *backtracking loop*. Note that, based on Lemmas 1 and 3, NMA is well defined for the proposed approach.

It is worth mentioning that the combination of a nonmonotone line search strategy and the BB step-size with safeguards was originally introduced by Rayden [37] for unconstrained optimization problems, and by Birgin et al. [9,10] for convex constrained optimization problems. In the new algorithm, we also somehow utilize this idea according to the above NMA technique.

4.3 New algorithm

Flow chart for GSCG in Fig. 1 is presented and then the GSCG algorithm is described.

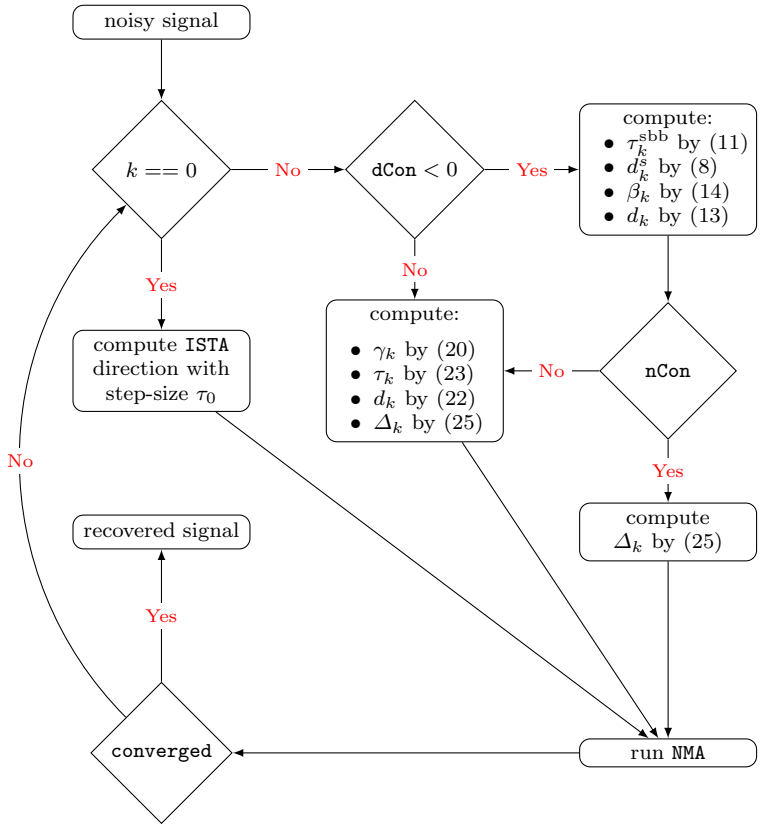


Fig. 1 Flow chart for GSCG

Algorithm 1: GSCG (generalized shrinkage conjugate gradient algorithm)

Input: An initial point $x_0 \in \mathbb{R}^n$, $\tau_0 > 0$, $N > 0$, $s > 0$, $\varrho, \lambda, \omega, \sigma, \upsilon \in (0, 1)$, $\mu > 0$, $\eta_0 \in [\eta_{\min}, \eta_{\max}]$, $m(0) := 0$, $0 < \eta_{\min} < \eta_{\max} < 1$ and $0 < \tau_{\min} < \tau_{\max} < \infty$.

```
1  begin
2      for k = 0, 1, 2, ... do
3          if k > 0 then
4              compute dCon by (16);
5              if dCon then
6                  set τ_k := τ_k^sbb, compute d_k^s by (8), β_k by (14) and d_k by (13);
7                  if nCon then
8                      compute Δ_k by (25);
9                  else
10                     compute γ_k by (20), τ_k by (23), d_k by (22) and Δ_k by (25);
11                 end
12             else
13                 compute γ_k by (20), τ_k by (23), d_k by (22) and Δ_k by (25);
14             end
15         else
16             set τ_k := τ_0 and compute d_k by (22) and Δ_k by (25);
17         end
18         run [x_{k+1}, F_{k+1}, ∇f_{k+1}, η_{k+1}, m(k+1)] =
           NMA(x_k, d_k, σ, ϱ, Δ_k, η_k, m(k), N);
19         if converged, break; end
20         s_k := x_{k+1} − x_k; y_k := ∇f_{k+1} − ∇f_k;
21         compute τ_{k+1}^sbb by (11);
22         k ← k + 1;
23     end
24 end
```

Output: $x^* := x_k$; $F^* := F_k$.

In Algorithm 1, if (16) or (21) is not satisfied (cGrad $= 0$) then to compute d_k, GSCG uses the CG step-size (Lines 10 and 13). Otherwise (cGrad $= 1$), it uses the BB step-size (Line 6). In addition, Lines 7–11 help us to establish the convergence rate.

5 Convergence analysis

This section is devoted to analyzing the convergence of GSCG. To do so, we use some tools for nonsmooth and smooth optimization which have been presented in [4,24,25,30,38], but have been extensively modified to allow for GSCG. In order to investigate convergence analysis, we utilize the following two assumptions:

(H1) The level set $L(x_0) := \{x \mid F(x) \leq F_0\}$ is bounded, for any $x_0 \in \mathbb{R}^n$.
(H2) The $\nabla f(x)$ is Lipschitz continuous with constant L.

The following lemma characterizes a stationary point. Our approach in it follows that of Lemma 2 in [38].

Lemma 4 *Suppose that $\tau_k > 0$, $\beta_k \geq 0$ and d_k is a direction generated by* GSCG. *Then, x_k is a stationary point for the problem* (1) *if and only if $d_k = 0$.*

Proof If (16) or (21) is not satisfied, then $d_k = d_k^s$. Thus, Lemma 2 gives the desired result. Otherwise, $d_k = d_k^s + \beta_k d_{k-1}$. In the case $d_k \neq 0$, Lemma 3 implies that $F'(x_k; d_k) < 0$, thus x_k is not a stationary point. Now, take $d_k = 0$. Since $d_k^s = -\beta_k d_{k-1}$ is the solution of (7), the descent condition (16) results in

$$\begin{aligned}
&\alpha\langle\nabla f_k, d\rangle + \frac{\alpha^2}{2\tau_k}\|d\|^2 + \mu\|x_k + \alpha d\|_1 \\
&\geq -\beta_k\langle\nabla f_k, d_{k-1}\rangle + \frac{\beta_k^2}{2\tau_k}\|d_{k-1}\|^2 + \mu\|x_k - \beta_k d_{k-1}\|_1 \\
&\geq -\beta_k\Big(\langle\nabla f_k, d_{k-1}\rangle + \mu\|d_{k-1}\|_1\Big) + \mu\|x_k\|_1 \\
&\geq \mu\|x_k\|_1,
\end{aligned}$$

for any $\alpha d \in \mathbb{R}^n$ with $\alpha > 0$. Hence, proceeding the same argument as Lemma 2 in [38], we get $F'(x_k; d) \geq 0$, leading to this fact that x_k is a stationary point of $F(x)$. □

In the sequel, let us define the following index sets:

$$\begin{aligned}
\mathcal{I}_1 &:= \Big\{k \mid d_k = d_k^s + \beta_k d_{k-1} \ (\text{or } \tau_k = \tau_k^{\text{sbb}})\Big\} \quad \text{and} \\
\mathcal{I}_2 &:= \Big\{k \mid d_k = d_k^s \ (\text{or } \tau_k = \tau_k^{\text{sbb}} + \omega\gamma_k\tau_{k-1})\Big\};
\end{aligned}$$

$\mathcal{I}_1$ is the set of all iterations where `cGrad=1` and $\mathcal{I}_2$ includes all iterations where `cGrad` $= 0$.

Lemma 5 *Let $\{\tau_k\}_{k\geq 0}$ be the sequence generated by* GSCG. *Then, $\{\tau_k\}_{k\geq 0}$ is bounded.*

Proof The proof is done in two cases:

(i) $k \in \mathcal{I}_1$. Then, relation (11) gives the result.
(ii) $k \in \mathcal{I}_2$. In this case, the proof is by induction. Since $\lim_{k\to\infty}\gamma_k = 0$, there exists $m \in \mathbb{N}$ such that $\gamma_k \leq \frac{1}{2}$, for all $k \geq m$. Let $T := \max\{\tau_{\max}, \tau_m\}$; it is clear that $\tau_m \leq 2T$. Suppose that $\tau_k \leq 2T$ for some $k \geq m$. Then, by (19) and the induction hypothesis, for all $k \geq m$, we have

$$\tau_{k+1} \leq \tau_{\max} + \omega\gamma_{k+1}\tau_k \leq T + \frac{1}{2}\tau_k \leq 2T,$$

which results in the proof. □

Lemma 6 *Suppose that the sequence* $\{x_k\}_{k\geq 0}$ *is generated by* GSCG *and Assumptions (H1) and (H2) hold. Then, there exists a constant* $\overline{\alpha} \in (0, 1]$ *such that the acceptance criterion* (26) *is satisfied for any* $\alpha \in (0, \overline{\alpha}]$. *In addition, for all* α_k *that satisfies* (26), *we can find a lower bound* $\underline{\alpha}$ *such that* $\alpha_k \geq \underline{\alpha}$.

Proof Lipcshitz continuity of ∇f, convexity of $\|\cdot\|_1$ and the definition of Δ_k in (25) result in

$$\begin{aligned} F(x_k + \alpha d_k) - R_k &\leq F(x_k + \alpha d_k) - F_k \\ &\leq \langle \nabla f_k, \alpha d_k \rangle + \frac{\alpha^2}{2} L \|d_k\|^2 + \alpha\mu\Big(\|x_k + d_k\|_1 - \|x_k\|_1\Big) \\ &= \alpha\Delta_k + \frac{\alpha^2}{2} L\|d_k\|^2. \end{aligned}$$

From (24), it follows that

$$F(x_k + \alpha d_k) - R_k \leq \alpha(1 - \alpha L\tau_k)\Delta_k. \tag{29}$$

Therefore, the acceptance criterion (26) is satisfied whenever

$$\alpha\Big(1 - \alpha L\tau_k\Big)\Delta_k \leq \sigma\alpha\Delta_k. \tag{30}$$

Since x_k is not stationary, based on Lemma 4 and (24), $\Delta_k < 0$. Thus, (30) yields to

$$\alpha \leq \overline{\alpha},$$

where $\overline{\alpha} := \dfrac{1-\sigma}{\overline{\tau}_{\max} L}$ with

$$\overline{\tau}_{\max} := \max\left\{\max_{k\in\mathcal{I}_1} \tau_k, \max_{k\in\mathcal{I}_2} \tau_k\right\} = \max\left\{\tau_{\max}, \max_{k\in\mathcal{I}_2} \tau_k\right\}.$$

Proving the lower bound for α_k, we know that either $\alpha_k = 1$ or the acceptance criterion (26) will fail at least once; hence

$$F\left(x_k + \frac{\alpha_k}{\varrho} d_k\right) > R_k + \sigma\frac{\alpha_k}{\varrho}\Delta_k.$$

This fact along with (29) leads to

$$\alpha_k - \frac{\varrho\Big(1-\sigma\Big)}{\tau_k L} > 0,$$

so that

$$\alpha_k > \underline{\alpha} := \frac{\varrho\left(1-\sigma\right)}{\overline{\tau}_{\max} L},$$

since $\overline{\tau}_{\max} \geq \tau_k$ for all k. Thus, the proof is completed. □

The proof of the next lemma has been inspired by the works of [4,24] who analyzed a nonmonotone line search for smooth optimization problems, but it has been broadly modified to allow for the problem (1) with the acceptance criterion (26) which is analogous to that of ibid.

Lemma 7 *Suppose that the sequence $\{x_k\}_{k\geq 0}$ is generated by* GSCG *with acceptance criterion (26) and Assumptions (H1) and (H2) hold. Then,*

(a) the sequence $\{F_{l(k)}\}_{k\geq 0}$ is convergent.
(b) $\lim_{k\to\infty} d_k = 0$.
(c) $\lim_{k\to\infty} F_k = \lim_{k\to\infty} F_{l(k)}$.
(d) $\lim_{k\to\infty} R_k = \lim_{k\to\infty} F_k$.

Proof (a) We show that the sequence $\{F_{l(k)}\}_{k\geq 0}$ is non-increasing. The inequality $R_k \leq F_{l(k)}$ implies that

$$F_{k+1} \leq R_k \leq F_{l(k)}. \tag{31}$$

For the case $k+1 \geq N$, from (31) and $m(k+1) := N-1$, we have

$$F_{l(k+1)} = \max_{0\leq j\leq N-1}\{F_{k-j+1}\} \leq \max_{0\leq j\leq m(k)+1}\{F_{k-j+1}\} = \max\{F_{l(k)}, F_{k+1}\} = F_{l(k)}.$$

Furthermore, for the case $k+1 < N$, we have $m(k+1) := k+1$ and $F_{l(k+1)} := F_0$. These cases give the non-increasing property of the sequence $\{F_{l(k)}\}_{k\geq 0}$. In addition,

$$F_{k+1} \leq R_k \leq F_{l(k)} \leq F_{l(k-1)} \leq \cdots \leq F_{l(0)} = F_0, \tag{32}$$

so that $\{x_k\}_{k\geq 0}$ remains in $L(x_0)$ for all k. Also, based on the Assumption (H1) and the definition of $F(x)$, $L(x_0)$ is compact. Thus, $\{F_{l(k)}\}_{k\geq 0}$ assumes a limit, named $\widetilde{F}$, for $k \to \infty$.

(b) Replacing k with $l(k)-1$ in (26) and using (31), we get

$$F_{l(k)} \leq F_{l(l(k)-1)} + \sigma\alpha_{l(k)-1}\Delta_{l(k)-1}, \tag{33}$$

which, along with part (a), implies that

$$\lim_{k\to\infty} \alpha_{l(k)-1}\Delta_{l(k)-1} = 0.$$

Hence, (24) gives

$$\lim_{k\to\infty} \alpha_{l(k)-1}\|d_{l(k)-1}\| = 0. \tag{34}$$

Based on Lemma 6, $\alpha_r \geq \underline{\alpha}$ for all r. Thus, (34) implies that

$$\lim_{k\to\infty} d_{l(k)-1} = 0. \tag{35}$$

Now, by induction, for any $j \geq 1$, we show that

$$\lim_{k\to\infty} d_{l(k)-j} = 0, \tag{36}$$

and

$$\lim_{k\to\infty} F_{l(k)-j} = \widetilde{F}. \tag{37}$$

It has been already shown in (35) that (36) holds for $j = 1$; hence

$$\lim_{k\to\infty} \|x_{l(k)} - x_{l(k)-1}\| = 0. \tag{38}$$

From (38) and the uniform continuity of $F(x)$ on $L(x_0)$, we get (37) for $j = 1$. Now, we assume that (36) and (37) hold for a given j. Setting $l(k) := l(k) - j$ in (33), we get

$$F_{l(k)-j} \leq F_{l(l(k)-(j+1))} + \sigma\alpha_{l(k)-(j+1)}\Delta_{l(k)-(j+1)},$$

where k is assumed to be large enough such that $l(k) - (j + 1) \geq 0$. By letting $k \to \infty$, using the inductive hypothesis along with part (a) and following the same arguments employed for driving (35), we deduce

$$\lim_{k\to\infty} d_{l(k)-(j+1)} = 0,$$

so that

$$\lim_{k\to\infty} \|x_{l(k)-j} - x_{l(k)-(j+1)}\| = 0.$$

This fact and the uniform continuity of $F(x)$ on $L(x_0)$ result in

$$\lim_{k\to\infty} F_{l(k)-(j+1)} = \lim_{k\to\infty} F_{l(k)-j} = \widetilde{F},$$

proving the inductive step. Based on (28), let $l(k) := \arg\max \left\{ F_j \mid \max(0, k - N+1) \leq j \leq k \right\}$. Thus, $l(k)$ is one of the members of the index set $\{k-N+1, k-N+2, \ldots, k\}$. Hence, we can denote $k-N = l(k)-j$, for some $j = 1, 2, \ldots, N$. Therefore, from (36) we deduce,

$$\lim_{k\to\infty} \alpha_k \|d_k\| = \lim_{k\to\infty} \alpha_{k-N} \|d_{k-N}\| = 0, \tag{39}$$

leading to

$$\lim_{k\to\infty} d_k = 0,$$

by Lemma 6.

(c) For any $k \in \mathbb{N}$, we have

$$x_{k-N} := x_{l(k)} - \sum_{j=1}^{l(k)-(k-N)} \alpha_{l(k)-j} d_{l(k)-j}. \tag{40}$$

Thus, (36) and (40) result in

$$\lim_{k\to\infty} \|x_{k-N} - x_{l(k)}\| = 0. \tag{41}$$

Part (a), along with (41) and the uniform continuity of $F(x)$ on $L(x_0)$, implies that

$$\lim_{k\to\infty} F_k = \widetilde{F}.$$

(d) From $F_k \le R_k \le F_{l(k)}$ and the previous part, we obtain the desired result. □

We now prove the main global convergence theorems of the new approach in two sections.

5.1 Convergence rate for convex case

In this part a sublinear convergence estimate for the error in the objective function value $F(x)$ is presented. The first theorem implies that GSCG converges to a global solution of the problem (1).

Theorem 1 *Suppose that the Assumptions (H1) and (H2) hold and let $\{x_k\}_{k\ge0}$ be the sequence generated by* GSCG. *Then, any accumulation point of $\{x_k\}_{k\ge0}$ is a stationary point of the problem* (1). *In addition,*

$$\lim_{k\to\infty} F_k = F^*,$$

where F^* is the optimal value for the problem (1).

Proof By (32), the sequence $\{x_k\}_{k\ge0}$ remains in $L(x_0)$. In addition, since $L(x_0)$ is compact, there exists an accumulation point. Furthermore, we get from (11) and Lemma 5

$$0 < \tau_{\min} \le \lim_{\substack{k\in\mathcal{I}_1\\ k\to\infty}} \tau_k = \lim_{\substack{k\in\mathcal{I}_2\\ k\to\infty}} \tau_k \le \tau_{\max} < +\infty. \tag{42}$$

On the other hand, the sequence $\{\nabla f_k\}_{k\ge0}$ is bounded since Assumption (H2) holds and $L(x_0)$ is compact. Lemma 7(b), (42), boundedness of $\{\nabla f_k\}_{k\ge0}$ and continuity of the shrinkage operator in (3) (see [27, Theorem 4.5]), result in

$$\lim_{\substack{k\in\mathcal{I}_1\\ k\to\infty}} d_k = \lim_{\substack{k\in\mathcal{I}_2\\ k\to\infty}} d_k = \lim_{k\to\infty} d_k^s(\tau_k^{\text{sbb}}),$$

where

$$d_k^s(\tau_k^{\text{sbb}}) =: \mathcal{S}_{\mu\tau_k^{\text{sbb}}}\left(x_k - \tau_k^{\text{sbb}}\nabla f_k\right) - x_k.$$

We consider the proof by contradiction. Let x^* be the accumulation point of the sequence $\{x_k\}_{k\geq 0}$ that is not stationary. Hence, by Lemma 4, for all sufficiently large k, there is an $\epsilon > 0$ such that $\|d_k\| \geq \epsilon > 0$. Thus, (39) results in

$$\lim_{k\to\infty} \alpha_k = 0,$$

which contradicts Lemma 6. By Lemma 7, $\{F_k\}_{k\geq 0}$ approaches a limit denoted by $\widetilde{F}$. Furthermore, from convexity of $F(x)$, a stationary point is a global minimizer; hence $\widetilde{F} = F^*$ which completes the proof. □

In the sequel, we show the sublinear convergence of GSCG in a similar way as Theorem 3.2 in [25] and Theorem 2 in [30].

Theorem 2 *Suppose that $\{x_k\}_{k\geq 0}$ is a sequence generated by* GSCG*, and Assumptions (H1) and (H2) hold. Then, there exists a constant c such that for all sufficiently large k,*

$$F_k - F^* \leq \frac{c}{k}.$$

Proof Convexity of $F(x)$ and $\alpha_k \in (0, 1]$ result in

$$F_{k+1} \leq (1-\alpha_k)F_k + \alpha_k F(x_k + d_k).$$

If (16) and (21) are satisfied, then $d_k = d_k^s + \beta_k d_{k-1}$. Taking

$$Q_{\tau_k}(x_k, d_k^s) := f_k + \langle \nabla f_k, d_k^s\rangle + \frac{1}{2\tau_k}\|d_k^s\|^2 + \mu\|x_k + d_k^s\|_1$$

and using Lipschitz continuity of ∇f, we get

$$\begin{aligned} F(x_k + d_k) &\leq f_k + \langle \nabla f_k, d_k^s\rangle + \frac{1}{2\tau_k}\|d_k^s\|^2 + \mu\|x_k + d_k^s\|_1 \\ &\quad + \beta_k\left(\langle \nabla f_k, d_{k-1}\rangle + \mu\|d_{k-1}\|_1\right) + \frac{L}{2}\|d_k\|^2 \\ &\leq Q_{\tau_k}(x_k, d_k^s) + \frac{L}{2}\|d_k\|^2. \end{aligned} \tag{43}$$

Otherwise, $d_k = d_k^s$. Again Lipschitz continuity of ∇f results in

$$\begin{aligned} F(x_k + d_k) &\leq f_k + \langle \nabla f_k, d_k^s \rangle + \frac{1}{2\tau_k}\|d_k^s\|^2 + \mu\|x_k + d_k^s\|_1 + \frac{L}{2}\|d_k\|^2 \\ &\leq Q_{\tau_k}(x_k, d_k^s) + \frac{L}{2}\|d_k\|^2. \end{aligned} \tag{44}$$

Since d_k^s is the minimizer of (7) and $f(x)$ is convex, for $\tau_k > 0$, it follows that

$$\begin{aligned} Q_{\tau_k}(x_k, d_k^s) &\leq \min_d \left\{ f_k + \langle \nabla f_k, d \rangle + \mu\|x_k + d\|_1 + \frac{1}{2\tau_k}\|d\|^2 \right\} \\ &\leq \min_d \left\{ F(x_k + d) + \frac{1}{2\tau_{\min}}\|d\|^2 \right\}. \end{aligned} \tag{45}$$

Let us now denote $x_k + d := (1-\vartheta)x_k + \vartheta x^*$, where $\vartheta \in [0, 1]$ and x^* is an optimal solution of the problem (1). Convexity of $F(x)$ yields to

$$\begin{aligned} \min_d \left\{ F(x_k + d) + \frac{1}{2\tau_{\min}}\|d\|^2 \right\} &\leq F\Big((1-\vartheta)x_k + \vartheta x^*\Big) \\ &\quad + \frac{1}{2\tau_{\min}}\|(1-\vartheta)x_k + \vartheta x^* - x_k\|^2 \\ &\leq (1-\vartheta)F_k + \vartheta F^* + \vartheta^2 \Phi_k, \end{aligned} \tag{46}$$

where $\Phi_k := \dfrac{1}{2\tau_{\min}}\|x_k - x^*\|^2$. By (32) and Assumption (H1), x_k and x^* lie in $L(x_0)$, so that $\Phi_k \leq c_1 < \infty$, where $c_1 := \dfrac{1}{2\tau_{\min}} D^2_{L(x_0)}$. We now get from (43) to (46)

$$F_{k+1} \leq (1 - \alpha_k\vartheta)F_k + \alpha_k(\vartheta F^* + c_1\vartheta^2) + \alpha_k\frac{L}{2}\|d_k\|^2. \tag{47}$$

Acceptance criterion (26) along with (24) results in

$$\frac{L\alpha_k}{2}\|d_k\|^2 \leq \frac{L\tau_k}{\sigma}\big(R_k - F_{k+1}\big) \leq c_2\big(R_k - F_{k+1}\big), \tag{48}$$

where $c_2 := \dfrac{L\overline{\tau}_{\max}}{\sigma}$. Combining (47), (48) and $F_k \leq R_k$, it follows that

$$F_{k+1} \leq R_k + \alpha_k(\vartheta F^* - \vartheta R_k + c_1\vartheta^2) + c_2\big(R_k - F_{k+1}\big). \tag{49}$$

The right hand side of (49) reaches its minimum value when $\vartheta_{\min} := \min\left\{1, \dfrac{R_k - F^*}{2c_1}\right\}$. As a consequence of Theorem 1 and Lemma 7, the sequence

$\{R_k\}_{k\geq 0}$ converges to F^*. Hence, $\vartheta_{\min}$ also approaches zero as $k \to \infty$; hence there is an integer k_0 with $\vartheta_{\min} < 1$ for all $k > k_0$. Therefore

$$\begin{aligned} F_{k+1} &\leq R_k - \frac{\alpha_k}{4c_1}(R_k - F^*)^2 + c_2\big(R_k - F_{k+1}\big) \\ &\leq R_k - c_3(R_k - F^*)^2 + c_2\big(R_k - F_{k+1}\big), \end{aligned} \tag{50}$$

for all $k > k_0$, where $c_3 := \dfrac{1}{4c_1}\underline{\alpha}$, with $\underline{\alpha}$ given by Lemma 6. Let us define $E_k := F_k - F^*$ and $E_{r(k)} := R_k - F^*$. By subtracting F^* from the both side of (50), we have

$$E_{k+1} \leq E_{r(k)} - c_3 E_{r(k)}^2 + c_2\big(E_{r(k)} - E_{k+1}\big),$$

so that

$$E_{k+1} \leq E_{r(k)} - c_4 E_{r(k)}^2,$$

where $c_4 := \dfrac{c_3}{1 + c_2}$. We find after division by $E_{r(k)} \neq 0$ that

$$\frac{E_{k+1}}{E_{r(k)}} \leq 1 - c_4 E_{r(k)},$$

yielding to

$$\frac{1}{E_{k+1}} \geq \frac{1}{E_{r(k)}} + c_4 \geq \frac{1}{E_{l(k)}} + c_4, \tag{51}$$

since $E_{k+1} \leq E_{r(k)}$, $R_k \leq F_{l(k)}$ and $E_{l(k)} := F_{l(k)} - F^*$. An integer i_0 can be found such that $k_0 \in \Big((i_0 - 1)N, i_0 N - 1\Big]$. For all $k \in \Big((i-1)N, iN - 1\Big]$ with $i > i_0$, the definition of $l(k)$ in (28) results in $k - N + 1 \leq l(k) \leq k$. By applying (51) recursively, for all $i > i_0$, we have

$$\frac{1}{E_k} \geq \frac{1}{E_{l(k)}} \geq \frac{1}{E_{l(k-N)}} + c_4 \geq \frac{1}{E_{l(k-(i-i_0)N)}} + (i - i_0)c_4,$$

so that

$$E_k \leq \frac{E_{l(k-(i-i_0)N)}}{1 + (i - i_0)c_4 E_{l(k-(i-i_0)N)}} \leq \frac{1}{(i - i_0)c_4} \leq \frac{2}{ic_4} \leq \frac{2N}{kc_4},$$

for all $k \in \Big((i-1)N, iN - 1\Big]$ with $i > 2i_0$. Since there exists a finite number of integers $k \in [1, 2i_0 N]$, one finds c_5 satisfying $c_5 > \frac{2}{c_4}$ and

$$E_k \leq \frac{c_5 N}{k}, \quad \forall k \in [1, 2i_0 N].$$

By taking $c := c_5 N$, the proof is completed. □

5.2 Convergence rate for strongly convex case

This part is dedicated to the R-linear convergence rate of GSCG, when $F(x)$ satisfies

$$F(y) \geq F(x^*) + \xi \|y - x^*\|^2 \tag{52}$$

for all $y \in \mathbb{R}^n$, with $\xi > 0$. The relation (52) holds, when $f(x)$ is a strongly convex function and therefore x^* is a unique minimizer of $F(x)$. The proof of the following theorem is similar to those of theorem 4.1 in [25] and Theorem 3 in [30].

Theorem 3 *Suppose that $\{x_k\}_{k\geq 0}$ is a sequence generated by* GSCG*, and Assumptions (H1) and (H2) and (52) hold. Given c_2 and $\underline{\alpha}$ satisfying Theorem 2 and Lemma 7, there are constants ψ and $\varphi \in (0, 1)$ such that for all sufficiently large k,*

$$F_k - F^* \leq \psi \varphi^k (F_0 - F^*).$$

Proof We first show that there exists $\chi \in (0, 1)$ such that

$$F_{k+1} - F^* \leq \chi (F_{l(k)} - F^*). \tag{53}$$

Assume that ϖ satisfies

$$0 < \varpi < \min \left\{ \frac{2c_2}{\underline{\alpha}}, \frac{1}{L}, \frac{\xi \tau_{\min}}{L} \right\}. \tag{54}$$

We consider two cases:

(i) If $\|d_k\|^2 \geq \varpi (F_{l(k)} - F^*)$, then we get from (48)

$$\frac{\varpi \underline{\alpha}}{2} (F_{l(k)} - F^*) \leq \frac{1}{2} \underline{\alpha} \|d_k\|^2 \leq c_2 (F_{l(k)} - F_{k+1}),$$

so that

$$F_{k+1} - F^* \leq \chi (F_{l(k)} - F^*),$$

where $\chi := (1 - \frac{\varpi \underline{\alpha}}{2c_2}) \in (0, 1)$; it can be obtained by (54).

(ii) If $\|d_k\|^2 < \varpi (F_{l(k)} - F^*)$, then (52) results in

$$\Phi_k = \frac{1}{2\tau_{\min}} \|x_k - x^*\|^2 \leq \frac{1}{2\tau_{\min}\xi} (F_k - F^*) \leq c_6 (F_{l(k)} - F^*),$$

where $c_6 = \dfrac{1}{2\tau_{\min}\xi}$. We now get from (43) to (46),

$$\begin{aligned} F_{k+1} &\leq (1-\alpha_k\vartheta)F_k + \alpha_k(\vartheta F^* + \Phi_k\vartheta^2) + \frac{L\alpha_k}{2}\|d_k\|^2 \\ &\leq F_{l(k)} + \alpha_k\left(c_6\vartheta^2 - \vartheta + \frac{L\varpi}{2}\right)(F_{l(k)} - F^*), \end{aligned}$$

so that

$$F_{k+1} - F^* \leq \left[1 + \alpha_k\left(c_6\vartheta^2 - \vartheta + \frac{L\varpi}{2}\right)\right](F_{l(k)} - F^*), \qquad \forall\ \vartheta \in [0,1].$$

Let $\chi := 1 + \alpha_k(c_6\vartheta^2 - \vartheta + \frac{L\varpi}{2})$; its minimum value is at

$$\widehat{\vartheta} = \left\{1, \frac{1}{2c_6}\right\} \in [0,1]. \tag{55}$$

If $\widehat{\vartheta} = 1$, then (55) gives $c_6 < \frac{1}{2}$. Thus, $\varpi \leq \frac{1}{L}$, obtained from (54), leads to

$$\chi = 1 + \alpha_k\left(c_6 - 1 + \frac{\varpi L}{2}\right) \leq 1 - \frac{1-\varpi L}{2}\alpha_k < 1.$$

Otherwise, the inequality $\varpi < \dfrac{\xi\tau_{\min}}{L}$, obtained from (54), gives

$$\chi = 1 + \alpha_k\left(\frac{1}{4c_6} - \frac{1}{2c_6} + \frac{\varpi L}{2}\right) = 1 - \alpha_k\left(\frac{1}{4c_6} - \frac{\varpi L}{2}\right) < 1.$$

Therefore, (53) holds for all $k \geq 0$.

By replacing k with $l(k)-1$ in (53), setting $k \geq N$ and utilizing monotonicity of $F_{l(k)}$, we have

$$F_{l(k)} - F^* \leq \chi(F_{l(l(k)-1)} - F^*) \leq \chi(F_{l(k-N)} - F^*). \tag{56}$$

There exists $i \geq 1$ such that $k \in \big((i-1)N, iN-1\big]$ for any $k \geq N$. Applying (56) recursively gives

$$F_k - F^* \leq F_{l(k)} - F^* \leq \chi^{i-1}\big(F_{l(k-(i-1)N)} - F^*\big).$$

Since $F_{k+1} \leq F_{l(k)}$ and $l\big(k-(i-1)N\big) \in (0, N]$, we get

$$F_{l\left(k-(i-1)N\right)} \leq \max\{F_{l(0)}, F_{l(1)}, F_{l(2)}, \ldots, F_{l(N-1)}\} = F_0,$$

so that

$$F_k - F_* \leq \chi^{i-1}(F_1 - F^*) \leq \frac{1}{\chi} \sqrt[N]{\chi^k}(F_0 - F^*),$$

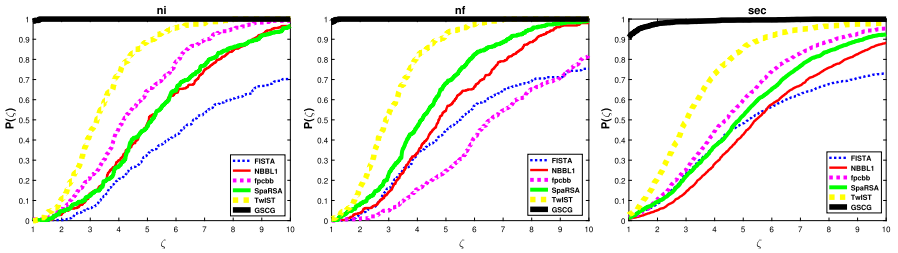
which completes the proof with $\psi = \frac{1}{\chi}$ and $\varphi = \sqrt[N]{\chi}$. □

6 Numerical results

We give some details of the implemented algorithms on CS problems in Sect. 6.1 and ID problems in Sect. 6.2.

Table 1 Amounts of parameters ρ and δ

ρ	δ	ρ	δ	ρ	δ
0.1	0.1	0.1	0.2	0.2	0.1
0.2	0.2	0.1	0.3	0.3	0.1
0.3	0.3	0.2	0.3	0.3	0.2



(a) (b) (c)

Fig. 2 A comparison among `TwIST`, `SpaRSA`, `fpcbb`, `NBBL1`, `FISTA` and `GSCG` with the performance measures `ni`, `nf` and `sec`, respectively

Table 2 Average of `ni`, `nf` and `sec` for all compared algorithms

Solver	ni	nf	sec
FISTA	2033	2034	114
fpcbb	3989	8674	315
NBBL1	1191	1375	88
SpaRSA	1164	1164	75
TwIST	735	825	41
GSCG	232	289	13

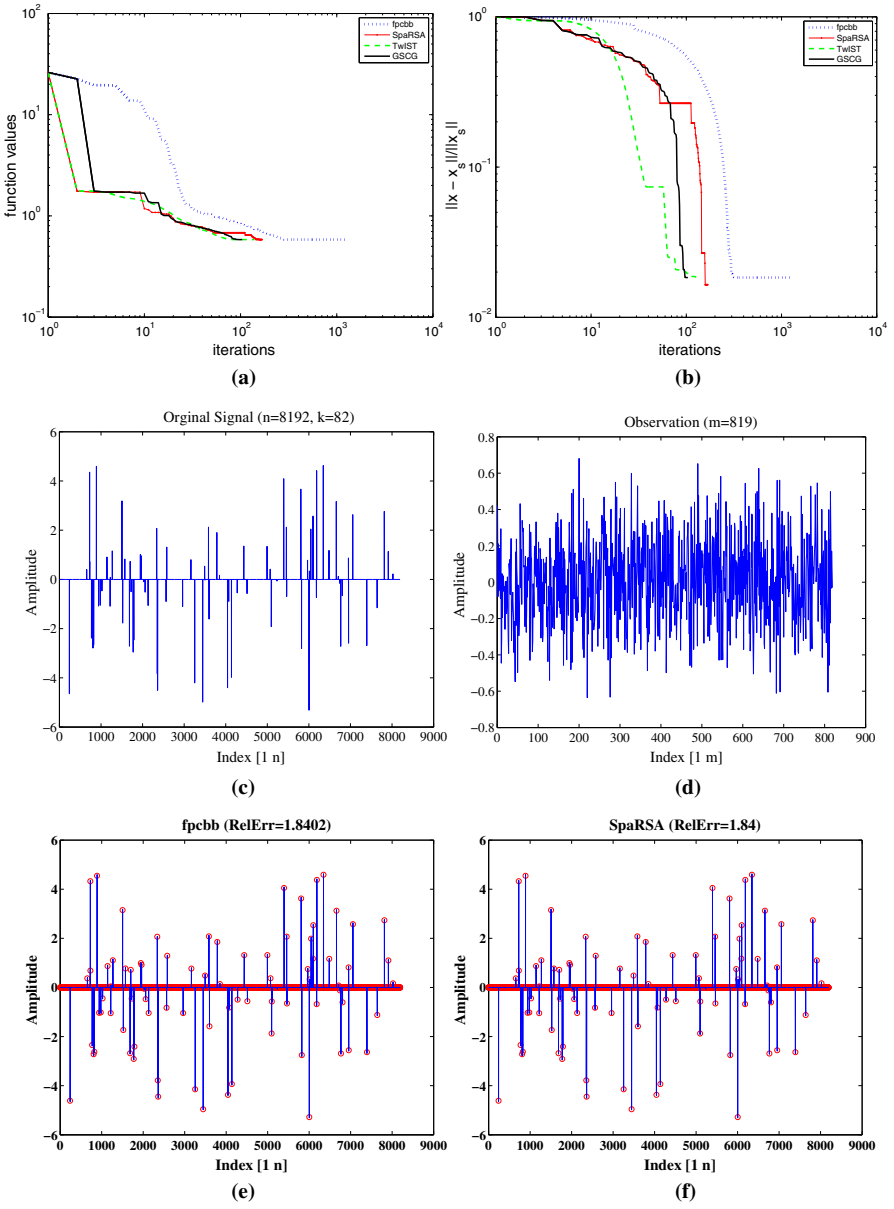


Fig. 3 A comparison among `fpcbb`, `SpaRSA`, `TwIST` and `GSCG` for matrix A in item (1) with $\sigma_1 = \sigma_2 = 10^{-7}$ and $\rho = \delta = 0.1$. **a** Diagram of function values versus iterations. **b** Diagram of real errors versus iterations. **c** Diagram of the original signal. **d** Diagram of the observation (noisy measurement). **e**–**h** Diagrams of recovered signals by `fpcbb`, `SpaRSA`, `TwIST` and `GSCG` (red circle) versus the original signal (blue peaks) (color figure online), respectively

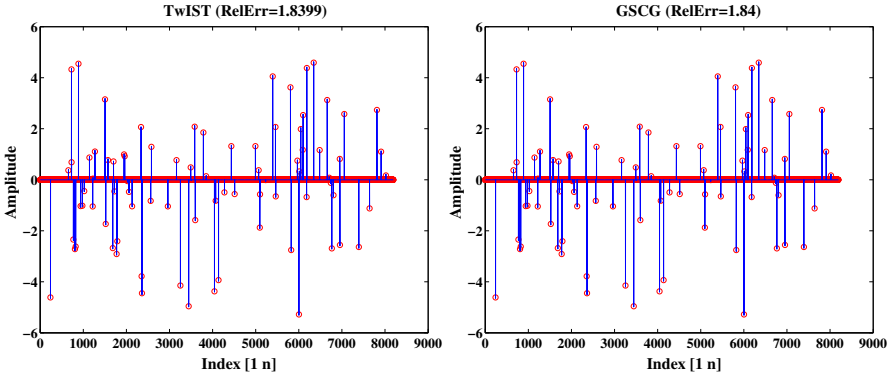


Fig. 3 continued

6.1 Quality of CS reconstruction

Here, the ability of GSCG for the sparse recovery in CS problems, a well-known application for (2), is evaluated. The field of CS, presented by Candès et al. [12] and Donoho [15], has grown considerably for the past few years. This fact that many real-world signals may be sparse or compressible in nature, well approximated by a sparse signal in a suitable basis or dictionary, is the motivation to create CS. Simply put it, CS refers to the idea of encoding a large sparse signal x through a relatively small number of linear measurements A and storing $b = Ax$, instead. The main aspect is decoding the observation vector b to recover the original signal x.

We report the results obtained by running our algorithm (GSCG) in comparison with fixed point continuation method with Barzilai–Borwein (fpcbb) [28], two-step ISTA (TwIST) [11], sparse reconstruction by separable approximation (SpaRSA) [41], nonmonotone BB gradient algorithm (NBBL1) [42] and fast iterative shrinkage-thresholding algorithm (FISTA) [8] on some CS problems. The codes of compared algorithms are available in

- (GSCG) https://github.com/GS1400/GSCGcode.git
- (fpcbb) http://www.caam.rice.edu/~optimization/L1/fpc/soft
- (TwIST) http://www.lx.it.pt/~bioucas/TwIST/TwIST.htm
- (SpaRSA) http://www.lx.it.pt/~mtf/SpaRSA
- (FISTA) http://iew3.technion.ac.il/~becka/papers/wavelet_FISTA.zip

Given the dimension of the signal n, according to the values of δ and ρ presenting in Table 1, we produce the dimension of the observation m and the number of nonzero elements k in an exact solution, by $m := \lfloor \delta n \rfloor$ and $k := \lfloor \rho m \rfloor$. Now we present six types of matrix A, see [27,28,39], used to produce our test problems as follows:

(1) Gaussian matrix (randn(m,n)) whose elements are pseudo random values drawn form the standard normal distribution $\mathcal{N}(0, 1)$;
(2) Scaled Gaussian matrix whose columns are scaled to unit norm;

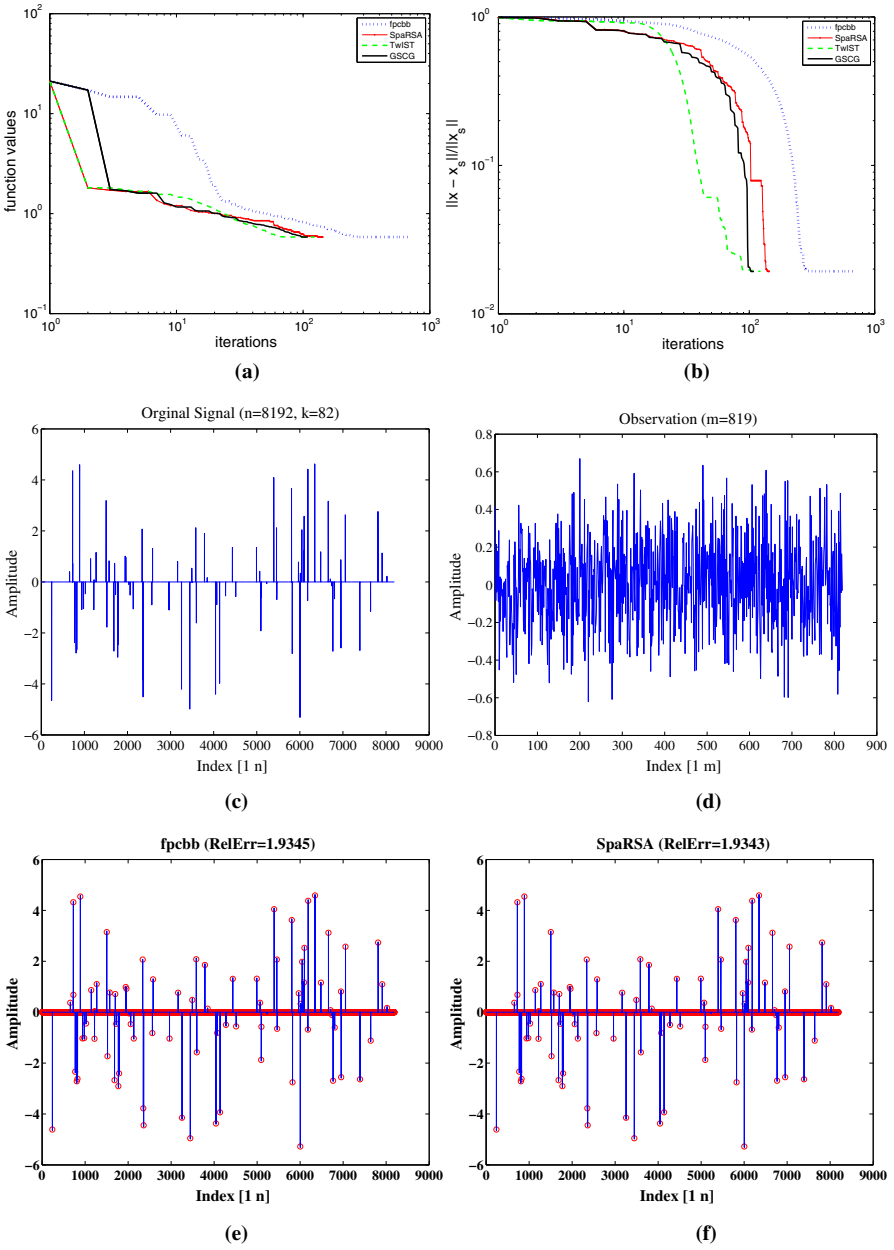


Fig. 4 A comparison among `fpcbb`, `SpaRSA`, `TwIST` and `GSCG` for matrix A in item (2) with $\sigma_1 = \sigma_2 = 10^{-7}$ and $\rho = \delta = 0.1$. **a** Diagram of function values versus iterations. **b** Diagram of real errors versus iterations. **c** Diagram of the original signal. **d** Diagram of the observation (noisy measurement). **e–h** Diagrams of recovered signals by `fpcbb`, `SpaRSA`, `TwIST` and `GSCG` (red circle) versus the original signal (blue peaks) (color figure online), respectively

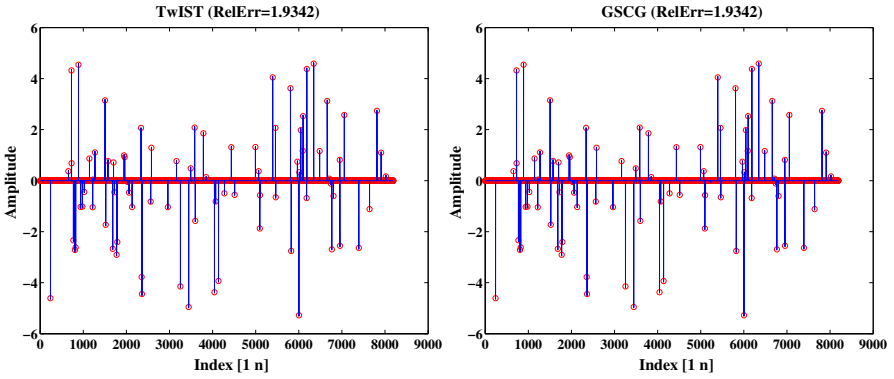


Fig. 4 continued

(3) Orthogonalized Gaussian matrix whose rows are orthogonalized by QR decomposition;
(4) Bernoulli matrix whose elements are $+/-1$ independently with equal probability;
(5) Partial Hadamard matrix, a matrix of $+/-1$ whose columns are orthogonal and whose m rows are chosen randomly from the $n \times n$ Hadamard matrix;
(6) Partial discrete cosine transform (PDCT) matrix whose m rows are chosen randomly from the $n \times n$ DCT matrix.

The matrices in items (1)–(5) are stored explicitly whose signals with dimensions $n \in \{2^{10}, \ldots, 2^{15}\}$ are tested. The matrices in item (6) are stored implicitly whose signals with dimensions $n \in \{2^{10}, \ldots, 2^{17}\}$ are tested. Since real problems are usually ruined by noise, according to the Gaussian noise in [27,28,39], we explain how to contaminate $\widetilde{x}$ and b by impulse noise in the following procedure:

Procedure(for producing $\widetilde{x}$ and b)

Input: n, k, σ_1, σ_2 and A
begin
  $x_s := \text{zeros}(n, 1)$;
  $p := \text{randperm}(n)$;
  $x_s(p(1:k)) := 2\text{randn}(k, 1)$;
  $\widetilde{x} := x_s + \sigma_1 \text{randn}(n, 1)$;
  $s_I := \text{randn}(m, 1)$;
  $b := A\widetilde{x} + \sigma_2 s_I$;
end
Output: $\widetilde{x}$ and b

In this procedure, the noise scenarios (σ_1, σ_2), which are input arguments, belong to

$$\left\{(10^{-1}, 10^{-1}), (10^{-3}, 10^{-3}), (10^{-5}, 10^{-5}), (10^{-7}, 10^{-7})\right\}.$$

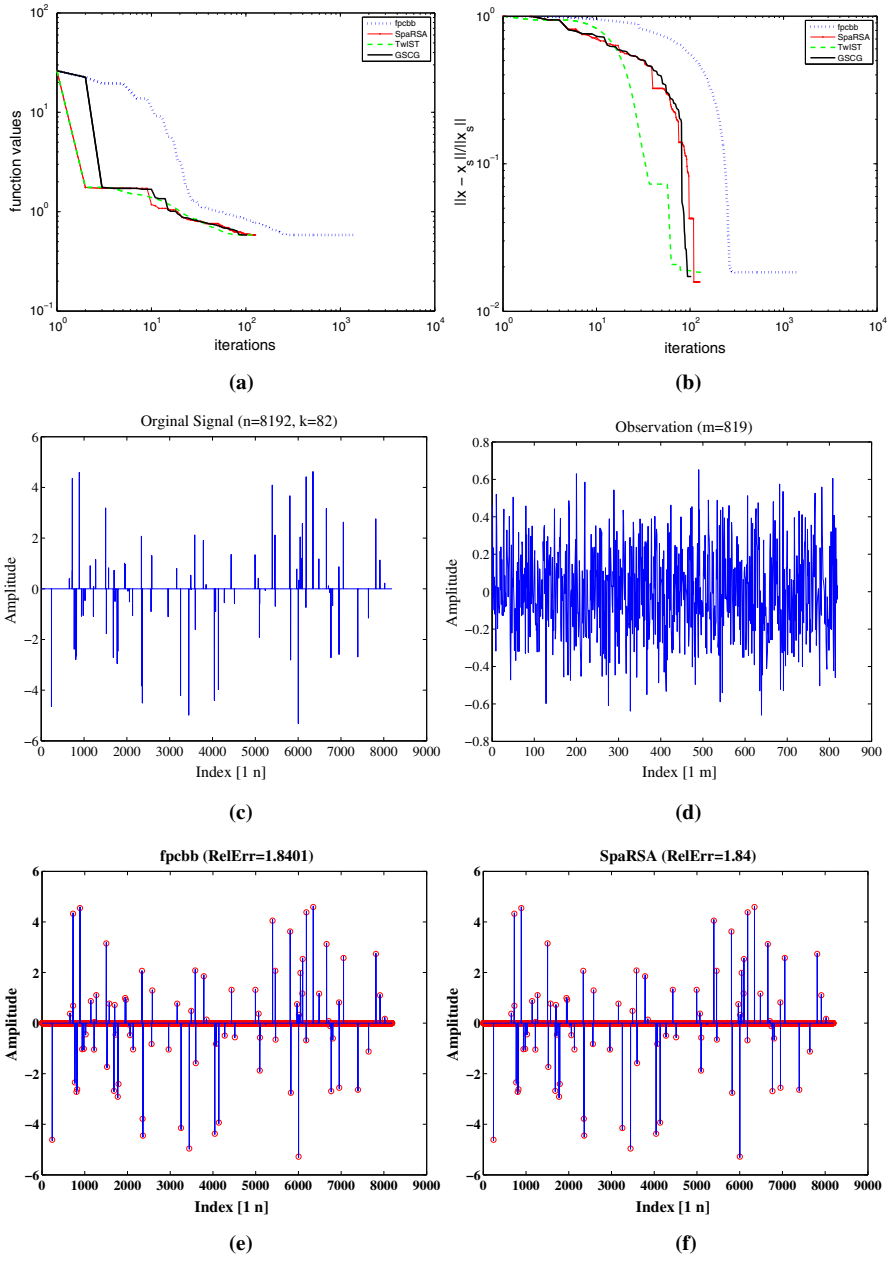


Fig. 5 A comparison among `fpcbb`, `SpaRSA`, `TwIST` and `GSCG` for matrix A in item (3) with $\sigma_1 = \sigma_2 = 10^{-7}$ and $\rho = \delta = 0.1$. **a** Diagram of function values versus iterations. **b** Diagram of real errors versus iterations. **c** Diagram of the original signal. **d** Diagram of the observation (noisy measurement). **e**–**h** Diagrams of recovered signals by `fpcbb`, `SpaRSA`, `TwIST` and `GSCG` (red circle) versus the original signal (blue peaks) (color figure online), respectively

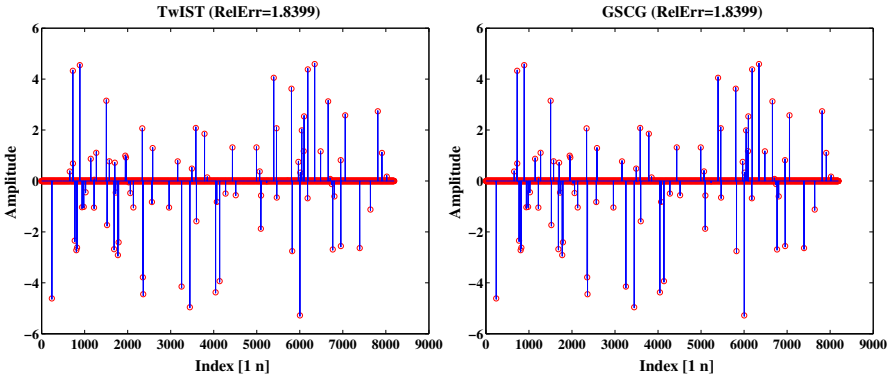


Fig. 5 continued

The n-by-1 matrix containing zeros is returned in Line 2 and a random permutation of the integers from 1 to n is returned in Line 3. Line 4 returns the original signal of the test problems which is an n-by-1 matrix containing k nonzero pseudo random values drawn from the standard normal distribution and $n - k$ zero elements. Both $\widetilde{x}$ and b, respectively, contaminated by impulsive noise are returned in Line 5 and 7.

Now, we explain how to choose the other parameters of all algorithms as follows:

- For `fpcbb` and `GSCG`, like that of [27,28,40], the parameter τ_0 is chosen by

$$\tau_0 := \min\{2.665 - 1.665m/n, 1.999\}.$$

- For all algorithms, the initial point is selected by $x_0 := \text{zeros}(n, 1)$ and we set $\mu = 2^{-8}$ and $\sigma = 10^{-3}$.
- For `fpcbb` and GSCG, according to [4], the parameter η_k is updated by

$$\eta_k := \begin{cases} \frac{2}{3}\eta_{k-1} + 0.01, & \text{if } \|\nabla f_k\| \leq 10^{-2}, \\ \max\{0.99\eta_{k-1}, 0.5\}, & \text{else}, \end{cases}$$

  for which $\eta_0 = 0.85$.
- `GSCG` employs $\alpha_0 = s = 1$, $\lambda = 0.25$, $\upsilon = 0.999$ and $\omega = 0.001$ and uses $\tau_k^{\text{sbb}} = \tau_k^{\text{sbb},1}$.
- For `GSCG`, `fpcbb`, `NBBL1` and `SpaRSA`, $\tau_{\min} = 10^{-4}$ and $\tau_{\max} = 10^4$ are selected.
- For `GSCG`, `fpcbb` and `NBBL1`, we set $\alpha_0 = s = 1$.

Note that each algorithm is run five times. This fact and different choices of the above listed parameters lead to generate more than 1620 random test problems. In our numerical experiments, the algorithms are stopped whenever

$$\|f_{k+1} - f_k\| < \texttt{ftol}\|f_k\|,$$

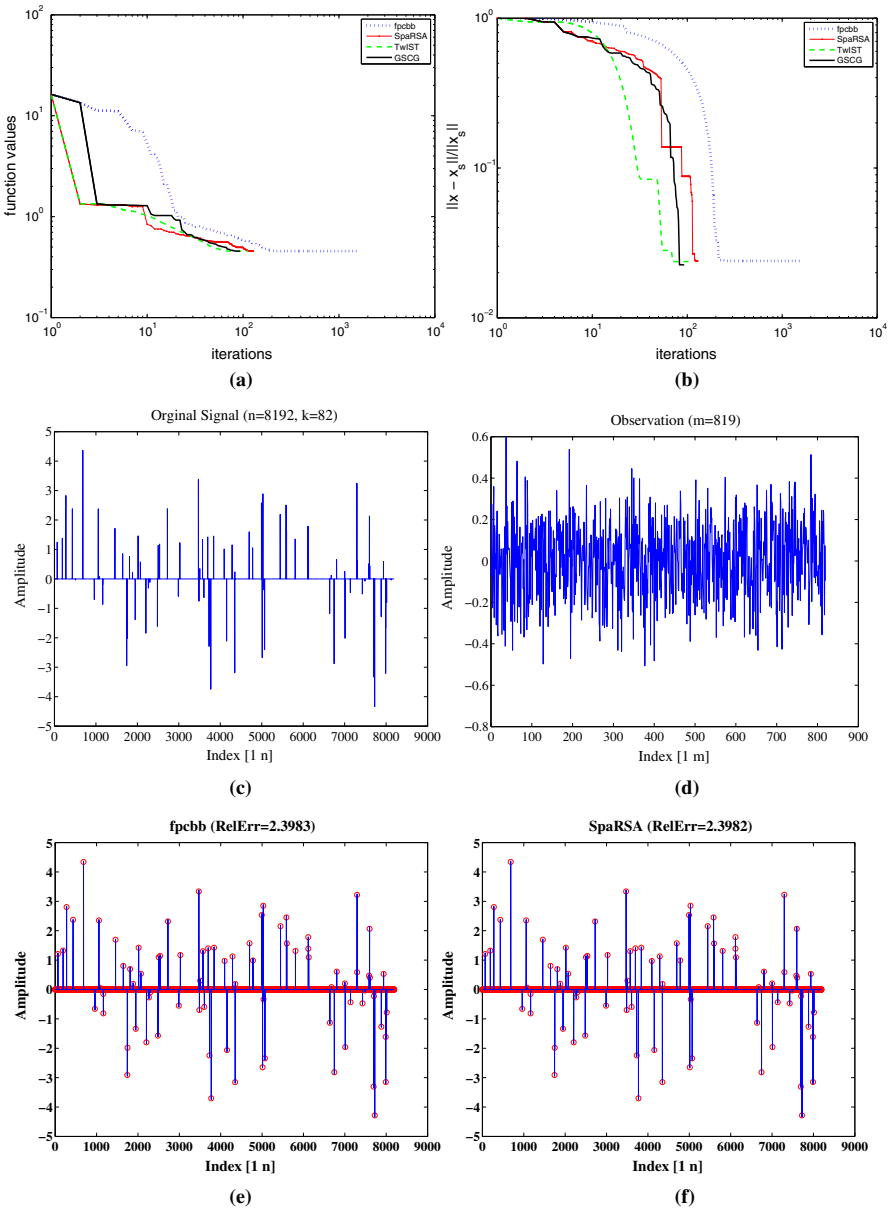


Fig. 6 A comparison among `fpcbb`, `SpaRSA`, `TwIST` and `GSCG` for matrix A in item (4) with $\sigma_1 = \sigma_2 = 10^{-7}$ and $\rho = \delta = 0.1$. **a** Diagram of function values versus iterations. **b** Diagram of real errors versus iterations. **c** Diagram of the original signal. **d** Diagram of the observation (noisy measurement). **e**–**h** Diagrams of recovered signals by `fpcbb`, `SpaRSA`, `TwIST` and `GSCG` (red circle) versus the original signal (blue peaks) (color figure online), respectively

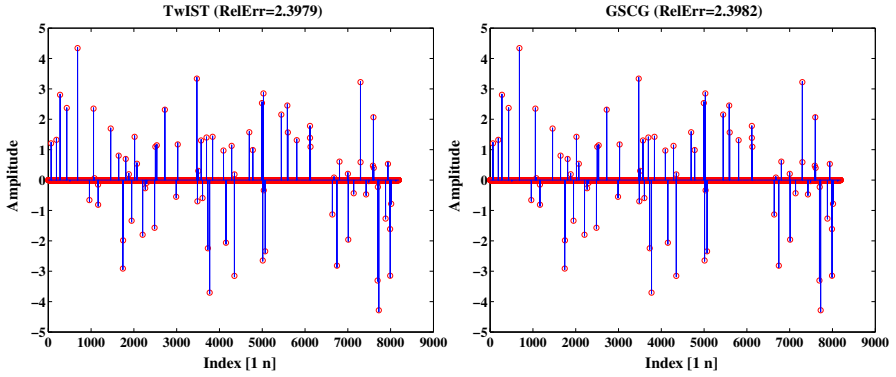


Fig. 6 continued

where ftol $:= 10^{-10}$ or the total number of iterations exceeds 10000.

Having a more reliable comparison, demonstrating the total behavior of the new presented procedure and better understanding of the performance of the all compared algorithms, we evaluated the performance profiles of all codes based on a set of metrics such as the number of total iterations (ni), the number of function evaluations (nf) and time in seconds (sec), from left to right in Fig. 2, by applying performance profiles MATLAB code proposed by Dolan and Morè in [14]. In subfigures a–c of Fig. 2, $P(\zeta)$ designates the percentage of problems which are solved within a factor ζ of the best solver.

In view of graphs depicted in the subfigures of Fig. 2, GSCG is clearly the winner of all performance metrics and attains most wins in terms of ni for around 99%, nf around 98% and sec around 92%. Furthermore, Fig. 2 verifies the ability of GSCG to solve the set of all test problems for $\zeta \geq 1$ in ni and nf and for $\zeta \geq 2$ in sec.

Table 2 contains the average of ni, nf and sec of the reconstructions with respect to the original signal x_s over 5 runs for the algorithms tested; these values are rounded (towards zero) to integers. These results show that, in solving problem (1), GSCG is faster than TwIST, SpaRSA and fpcbb and much faster than FISTA with the clear lowest values of ni and nf.

Let us first measure the quality of restoration x^* through the relative error to the original signal x_s by

$$\text{RelErr} := 100\frac{\|x^* - x_s\|}{\|x_s\|}.$$

Figures 3, 4, 5, 6, 7 and 8 are dedicated to the different six types of matrix A, respectively. These figures give some comparisons among GSCG, SpaRSA, TwIST and fpcbb in terms of function values versus iterations (subfigures a), relative error versus iterations (subfigures b) and the ability of reconstructions [subfigures c–h with recovered signal (red circles) versus original signal (blue peaks)]. These subfigures verify that GSCG is competitive with fpcbb, TwIST and SpaRSA in terms of com-

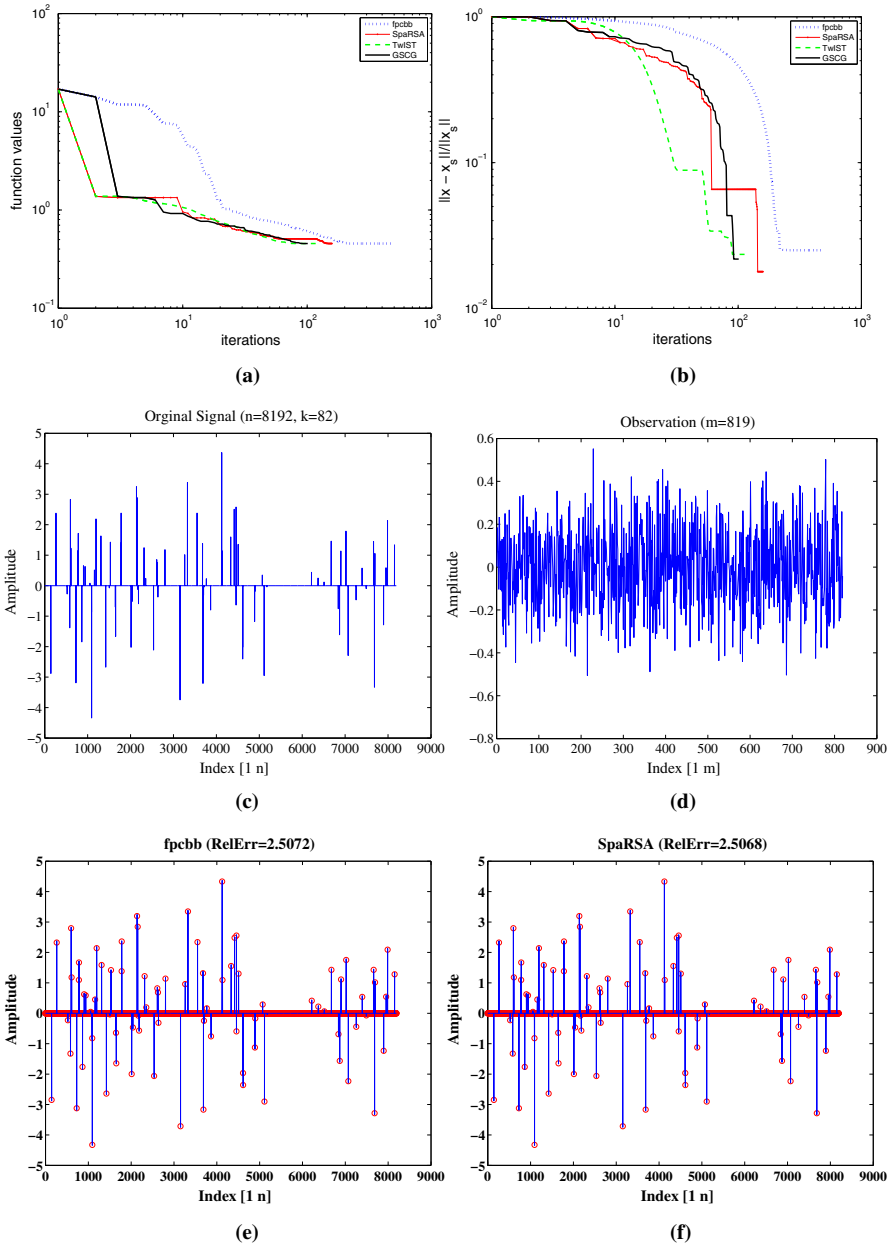


Fig. 7 A comparison among `fpcbb`, `SpaRSA`, `TwIST` and `GSCG` for matrix A in item (5) with $\sigma_1 = \sigma_2 = 10^{-7}$ and $\rho = \delta = 0.1$. **a** Diagram of function values versus iterations. **b** Diagram of real errors versus iterations. **c** Diagram of the original signal. **d** Diagram of the observation (noisy measurement). **e**–**h** Diagrams of recovered signals by `fpcbb`, `SpaRSA`, `TwIST` and `GSCG` (red circle) versus the original signal (blue peaks) (color figure online), respectively

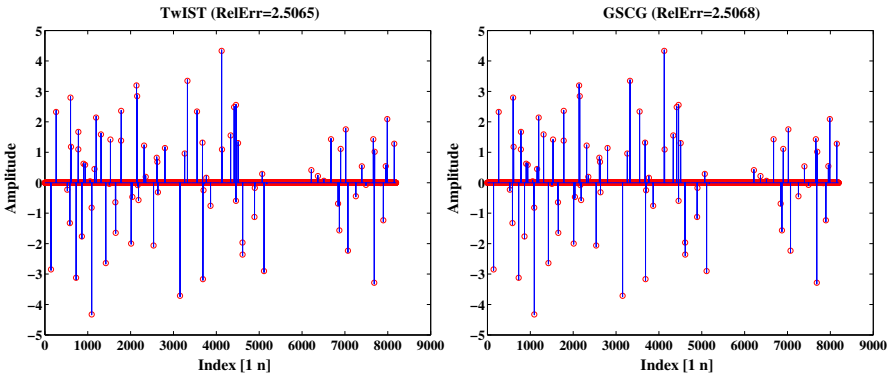


Fig. 7 continued

pared values. In addition, comparing graphs and noticing to the real errors of the reconstructions, one get the efficiency and robustness of the GSCG process in recovering large sparse signals in comparisons with the other solvers.

6.2 ID problem

Here, we present an application of the GSCG method to ID problems in order to demonstrate another applicative side of it. An image is a signal that conveys information about the behavior or attributes of a physical object. Often the image is more or less blurry which may arise due to environmental effects, sensor imperfection, communication errors or poor illumination. In ID, the main goal is to recover the original, sharp image. There are many directions and tools explored in studying ID. One of them is the class of algorithms which uses sparse and redundant representation modeling. In many situations, the blur is indeed linear or at least well approximated by a linear model which leads us to study the following linear system:

$$\begin{aligned} & b := Ax + \kappa, \\ & \text{s.t. } x \in X \end{aligned} \tag{57}$$

where X is a finite-dimensional vector space, x is a clean image, A is a blurring operator, b is the observed image and κ is an impulsive noise. Since (57) is mostly underdetermined and κ is not commonly available, an approximate solution of it can be found by solving (2). For a quantitative evaluation of the results, we use the peak signal-to-noise ratio PSNR defined as

$$\text{PSNR} = 20 \log_{10} \left(\frac{\sqrt{mn}}{\|x - x_t\|_F} \right) \text{ [dB]},$$

where $\|\cdot\|_F$ is the Frobenius norm, x_t denotes $m \times n$ true image and pixel values are in $[0, 1]$. In general, a higher PSNR points out that the reconstruction is of higher qual-

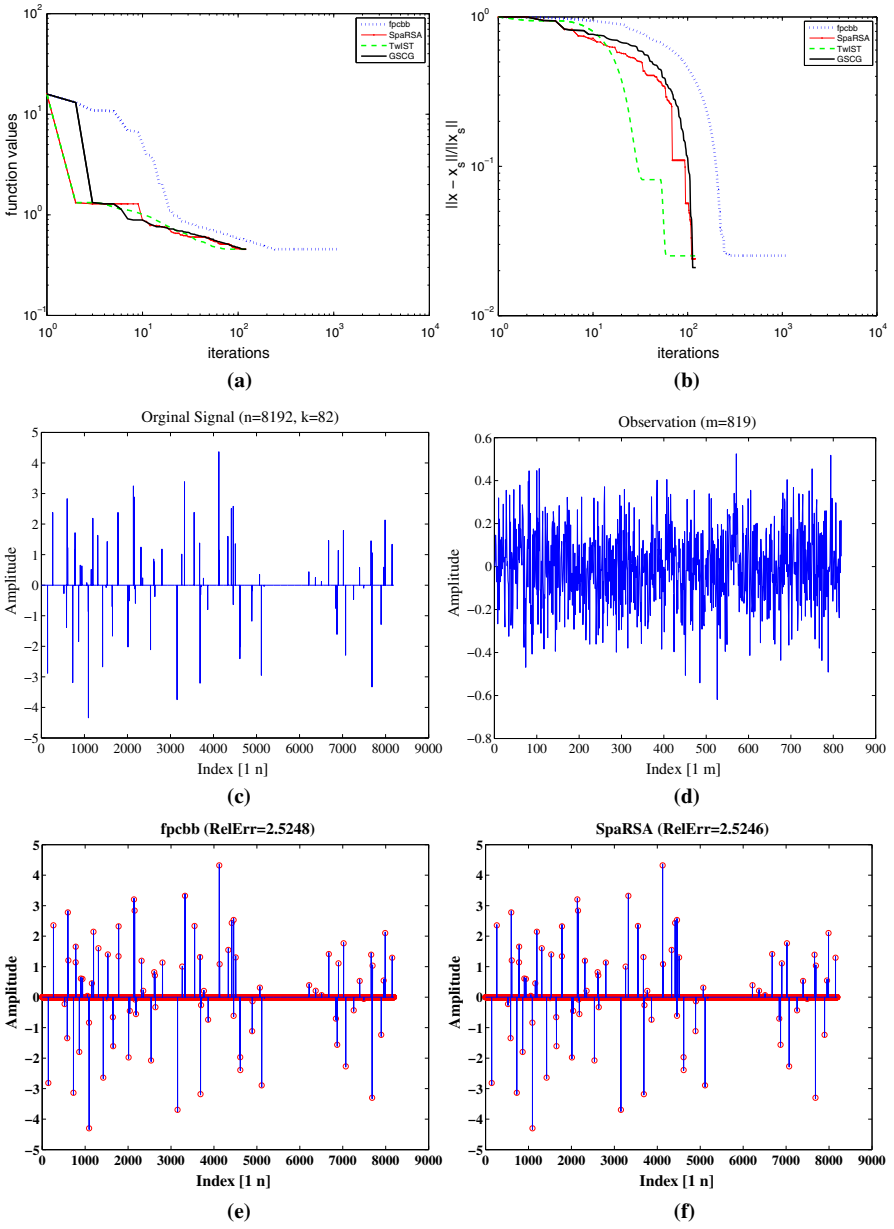


Fig. 8 A comparison among `fpcbb`, `SpaRSA`, `TwIST` and `GSCG` for matrix A in item (6) with $\sigma_1 = \sigma_2 = 10^{-7}$ and $\rho = \delta = 0.1$. **a** Diagram of function values versus iterations. **b** Diagram of real errors versus iterations. **c** Diagram of the original signal. **d** Diagram of the observation (noisy measurement). **e**–**h** Diagrams of recovered signals by `fpcbb`, `SpaRSA`, `TwIST` and `GSCG` (red circle) versus the original signal (blue peaks) (color figure online), respectively

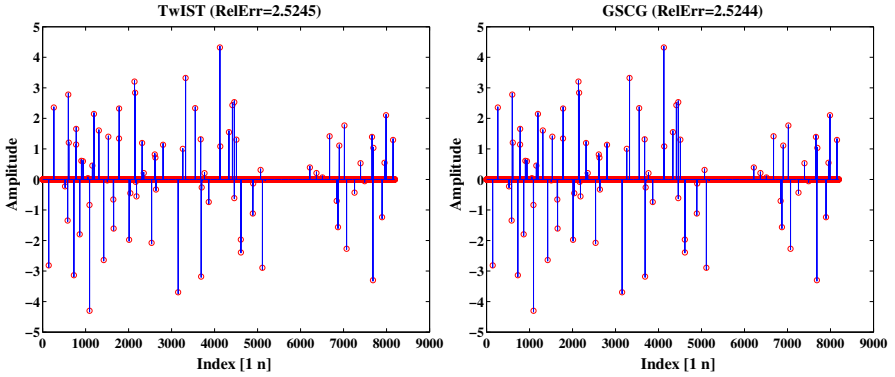


Fig. 8 continued

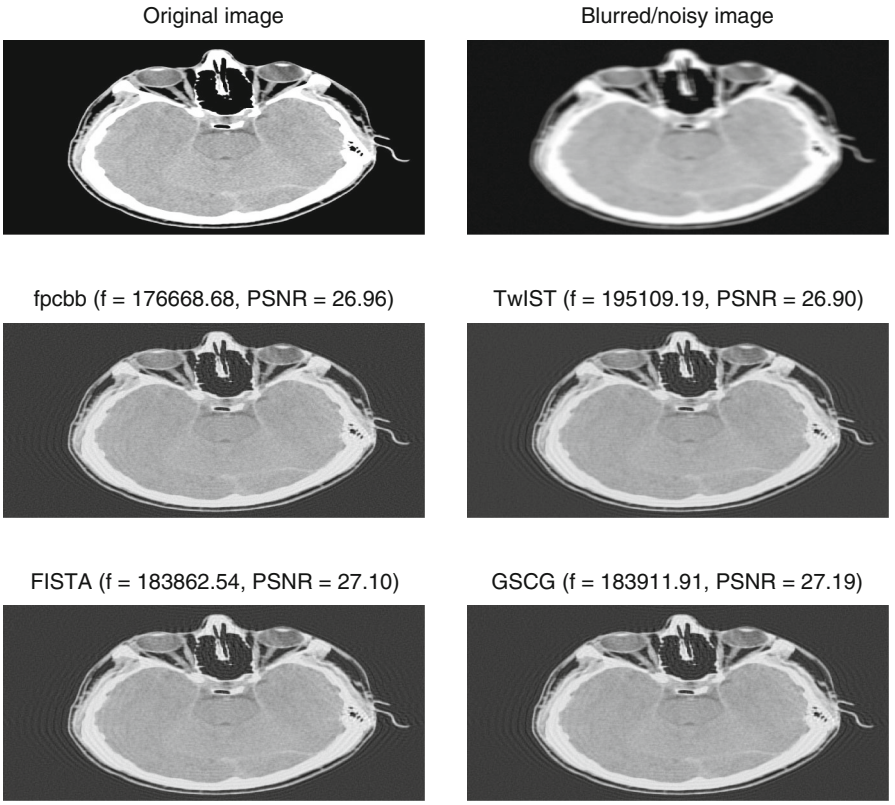


Fig. 9 Debluring the 512×512 HeadCT image with the 4×4 uniform blur and the Gaussian noise with `SNR` = 40 `dB` by `fpcbb`, `FISTA`, `TwIST` and `GSCG` with the regularization parameter $\mu := 5 \times 10^{-3}$. The algorithms were stopped after 50 iterations

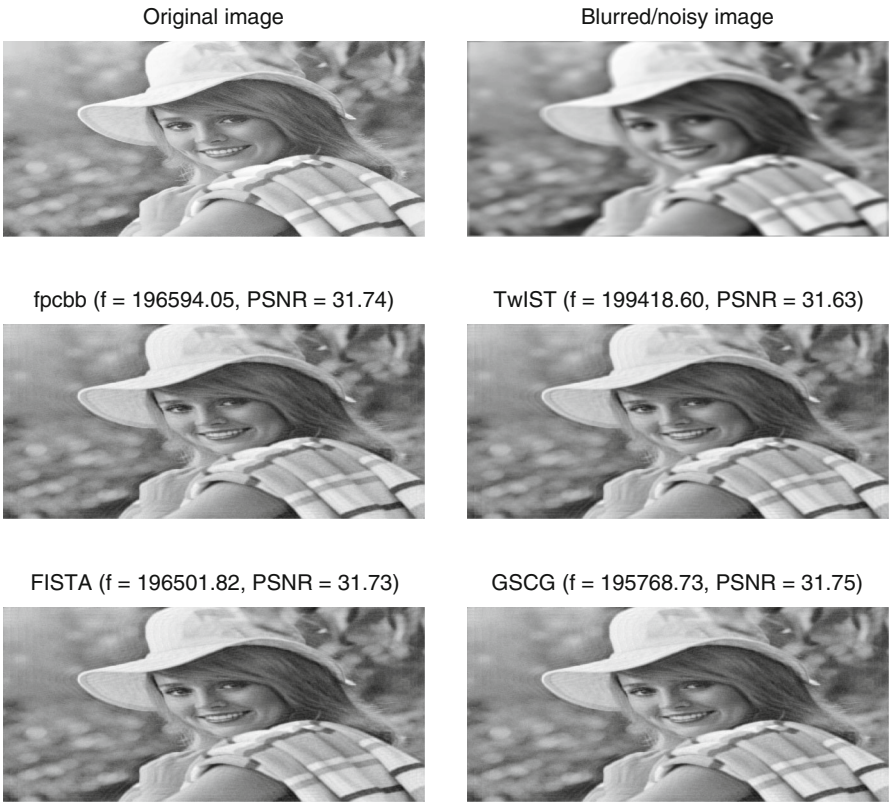


Fig. 10 Debluring the 512×512 Elaine image with the 4×4 uniform blur and the Gaussian noise with SNR = 40 dB by fpcbb, FISTA, TwIST and GSCG with the regularization parameter $\mu := 5 \times 10^{-3}$. The algorithms were stopped after 50 iterations

ity, see [1,2]. The true images are available in http://homepage.univie.ac.at/masoud.ahookhosh/uploads/OSGA_v1.1.tar.gz.

Here, the parameter values are taken the same as those in the CS experiment. In our numerical experiments, all algorithms are stopped whenever the total number of iterations exceeds 50. Figures 9, 10 and 11 show that GSCG achieves the best PSNR among fpcbb, FISTA and TwIST.

7 Conclusion

In this study, we have introduced and tested GSCG to solve the convex ℓ_1-regularized optimization problem. The main goal here is to improve and accelerate ISTA by introducing a new descent condition, a new search direction and a new shrinkage step-size. GSCG takes advantages of the new generated direction when it satisfies descent property under the new descent condition and also meets a mild norm condition. Otherwise, it utilizes the ISTA descent direction which uses the new shrinkage step-

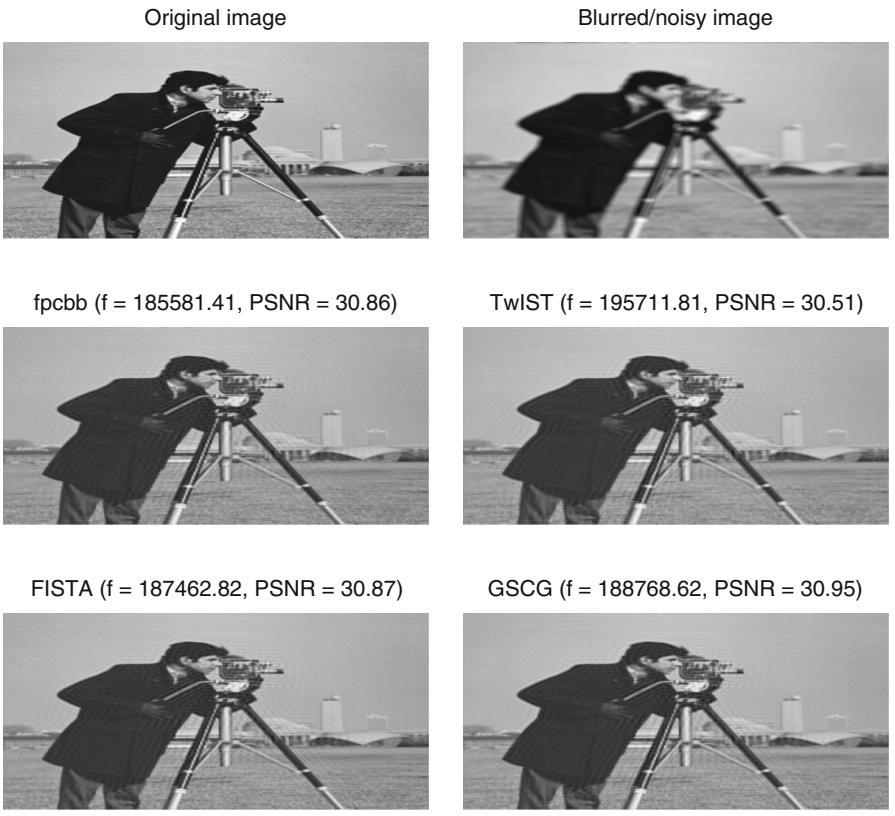


Fig. 11 Debluring the 512×512 Cameraman image with the 4×4 uniform blur and the Gaussian noise with SNR = 40 dB by fpcbb, FISTA, TwIST and GSCG with the regularization parameter $\mu := 5 \times 10^{-3}$. The algorithms were stopped after 50 iterations

size. Based on the CG idea, the new presented direction is a special linear combination of the ISTA descent direction and the previous direction. The new shrinkage step-size is a specific linear combination of the BB step-size and the previous step-size. The global convergence along with sublinear (R-linear) convergence rate of GSCG for the convex (strongly convex) ℓ_1-regularized optimization problem is established. In a series of numerical experiments, we give the experimental evidences that show the efficiency and robustness of GSCG in comparison with some state-of-the-art solvers when applied to the CS and ID problems.

Acknowledgements We would like to thank the high performance computing (HPC) center, a branch of institute for research in fundamental Physics and Mathematics (IPM), to help us to use HPC's cluster for computing numerical results. The third author acknowledges the financial support of the Doctoral Program "Vienna Graduate School on Computational Optimization" funded by Austrian Science Foundation under Project No. W1260-N35.

References

1. Ahookhosh, M.: High-dimensional nonsmooth convex optimization via optimal subgradient methods. Ph.D. Thesis, Faculty of Mathematics, University of Vienna (2015)
2. Ahookhosh, M.: User's manual for OSGA (Optimal SubGradient Algorithm). http://homepage.univie.ac.at/masoud.ahookhosh/uploads/User's_manual_for_OSGA.pdf (2014)
3. Ahokhosh, M., Amini, K.: An efficient nonmonotone trust-region method for unconstrained optimization. Numer. Algorithms **59**(4), 523–540 (2012)
4. Amini, K., Ahookhosh, M., Nosratpour, H.: An inexact line search approach using modified nonmonotone strategy for unconstrained optimization. Numer. Algorithm **66**, 49–78 (2014)
5. Amini, K., Shiker, M.A.K., Kimiaei, M.: A line search trust-region algorithm with nonmonotone adaptive radius for a system of nonlinear equations. 4OR-Q. J. Oper. Res. **14**(2), 133–152 (2016)
6. Barzilai, J., Borwein, J.M.: Two point step size gradient method. IMA J. Numer. Anal. **8**, 141–148 (1988)
7. Bazaraa, M.S., Sherali, S.D., Shetty, C.M.: Nonlinear Programming: Theory and Algorithms, 3rd edn. Wiley, New York (2006)
8. Beck, A., Teboulle, M.: A fast iterative shrinkage-thresholding algorithm for linear inverse problems. SIAM J. Imaging Sci. **2**(1), 183–202 (2009)
9. Birgin, E.G., Martìnez, J.M., Raydan, M.: Inexact spectral projected gradient methods on convex sets. IMA J. Numer. Anal. **23**(4), 539–559 (2003)
10. Birgin, E.G., Martìnez, J.M., Raydan, M.: Nonmonotone spectral projected gradient methods on convex sets. SIAM J. Optim. **10**(4), 1196–1211 (2000)
11. Bioucas-Dias, J., Figueiredo, M.: A new TwIST: two-step iterative shrinkage/thresholding algorithms for image restoration. IEEE Trans. Image Process. **16**, 2992–3004 (2007)
12. Candès, E., Romberg, J., Tao, T.: Robust uncertainty principles, exact signal reconstruction from highly incomplete frequency information. IEEE Trans. Inf. Theory. **52**(2), 489–509 (2006)
13. Combettes, P.L., Wajs, V.R.: Signal recovery by proximal forward–backward splitting. Multiscale Model. Simul. **4**(4), 1168–1200 (2005)
14. Dolan, E., Morè, J.J.: Benchmarking optimization software with performance profiles. Math. Program. **91**, 201–213 (2002)
15. Donoho, D.: Compressed sensing. IEEE Trans. Inf. Theory. **52**(4), 1289–1306 (2006)
16. Daubechies, I., Defrise, M., Mol, C.D.: An iterative thresholding algorithm for linear inverse problems with a sparsity constraint. Commun. Pure Appl. Math. **57**(11), 1413–1457 (2004)
17. Elad, M.: Sparse and Redundant Representation from Theory to Application in Signal and Image Processing. Springer, Berlin (2010). ISBN 978-1-4419-7011-4
18. Eldar, C.Y., Kutyniok, G.: Compressed Sensing: Theory and Application. Cambridge University Press, New York (2012). ISBN 978-1-107-00558-7
19. Esmaeili, H., Rostami, M., Kimiaei, M.: Combining line search and trust-region methods for ℓ_1-minimization. Int. J. Comput. Math. **95**(10), 1950–1972 (2018)
20. Figueiredo, M.A., Nowak, R.D.: An EM algorithm for wavelet-based image restoration. IEEE Trans. Image Process. **12**(8), 906–916 (2003)
21. Figueiredo, M.A., 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**(4), 586–597 (2007)
22. Fletcher, R., Reeves, C.: Function minimization by conjugate gradients. Comput. J. **7**, 149–154 (1964)
23. Foucart, S., Rauhut, H.: A Mathematical Introduction to Compressive Sensing. Springer, New York (2013)
24. Grippo, L., Lampariello, F., Lucidi, S.: A nonmonotone line search technique for Newton's method. SIAM J. Numer. Anal. **23**, 707–716 (1986)
25. Hager, W.W., Phan, D.T., Zhang, H.: Gradient based methods for sparse recovery. SIAM J. Imaging Sci. **41**, 146–165 (2011)
26. Hager, W.W., Zhang, H.: A survey of nonlinear conjugate gradient methods. Pac. J. Optim. **2**(1), 35–58 (2006)
27. Hale, E.T., Yin, W., Zhang, Y.: Fixed-point continuation for ℓ_1-minimization: methodology and convergence. SIAM J. Optim. **19**(3), 1107–1130 (2008)
28. Hale, E.T., Yin, W., Zhang, Y.: Fixed-point continuation applied to compressed sensing: implementation and numerical experiment. J. Comput. Math. **28**(2), 170–194 (2010)

29. Hestenes, M.R., Stiefel, E.L.: Methods of conjugate gradients for solving linear systems. J. Res. Nat. Bur. Stand. **49**, 409–436 (1952)
30. Huang, Y., Liu, H.: A Barzilai–Borwein type method for minimizing composite functions. Numer. Algorithm **69**, 819–838 (2015)
31. Iiduka, H.: Hybrid conjugate gradient method for a convex optimization problem over the fixed-point set of a nonexpansive mapping. J. Optim. Theory Appl. **140**, 463–475 (2009)
32. Iiduka, H., Yamada, I.: A use of conjugate gradient direction for the convex optimization problem over the fixed point set of a nonexpansive mapping. SIAM J. Optim. **19**(4), 1881–1893 (2009)
33. Kowalski, R.M.: Proximal algorithm meets a conjugate descent. Pac. J. Optim. **12**(3), 549–667 (2011)
34. Nocedal, J., Wright, S.: Numerical Optimization. Springer, Berlin (1999)
35. Parikh, N., Boyd, S.: Proximal algorithms. Found. Trend. Optim. **1**(3), 123–231 (2013)
36. Polak, E., Ribière, G.: Note sur la convergence de directions conjugées. Rev. Fr. Inform. Rech. Opèrtionnelle 3e Année **16**, 35–43 (1969)
37. Raydan, M.: The Barzilai and Borwein gradient method for the large scale unconstrained minimization problem. SIAM J. Optim. **7**(1), 26–33 (1997)
38. Tseng, P., Yun, S.: A coordinate gradient descent method for nonsmooth separable minimization. Math. Program. **117**(1), 387–423 (2009)
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**(4), 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. Methods Softw. **27**(6), 1127–1146 (2012)
41. Wright, S.J., Nowak, R.D., Figueiredo, M.A.T.: Sparse reconstruction by separable approximation. IEEE Trans. Signal Process. **57**(7), 2479–2493 (2009)
42. Xiao, Y., Wu, S.-Y., Qi, L.: Nonmonotone Barzilai–Borwein gradient algorithm for ℓ_1-regularized nonsmooth minimization in compressive sensing. J. Sci. Comput. **61**, 17–41 (2014)
43. Zhang, H.C., Hager, W.W.: A nonmonotone line search technique and its application to unconstrained optimization. SIAM J. Optim. **14**(4), 1043–1056 (2004)