Filomat 32:6 (2018), 2173–2191
https://doi.org/10.2298/FIL1806173E

Published by Faculty of Sciences and Mathematics,
University of Niš, Serbia
Available at: http://www.pmf.ni.ac.rs/filomat

Extended Dai–Yuan Conjugate Gradient Strategy for Large-Scale Unconstrained Optimization with Applications to Compressive Sensing

Hamid Esmaeili[a], Majid Rostami[a], Morteza Kimiaei[b]

[a] *Department of Mathematics, Bu-Ali Sina University, Hamedan, Iran*
[b] *Faculty of Mathematics, University of Vienna, Oskar-Morgenstern-Platz 1, A-1090 Vienna, Austria*

Abstract. We present a new spectral conjugate gradient method based on the Dai–Yuan strategy to solve large-scale unconstrained optimization problems with applications to compressive sensing. In our method, the numerator of conjugate gradient parameter is a convex combination from the maximum gradient norm value in some preceding iterates and the current gradient norm value. This combination will try to produce the larger step-size far away from the optimizer and the smaller step-size close to it. In addition, the spectral parameter guarantees the descent property of the new generated direction in each iterate. The global convergence results are established under some standard assumptions. Numerical results are reported which indicate the promising behavior of the new procedure to solve large-scale unconstrained optimization and compressive sensing problems.

1. Introduction

Consider the following unconstrained optimization problem

$$\begin{array}{ll} \min & f(x) \\ \text{s.t.} & x \in \mathbb{R}^n, \end{array} \tag{1}$$

for which $f : \mathbb{R}^n \longrightarrow \mathbb{R}$ is a continuously differentiable function and bounded from below. There are many kinds of iterative methods to solve unconstrained optimization problems including the Newton method [28], the steepest descent method [29], the conjugate gradient methods [11, 13, 16, 23], the quasi-Newton methods [9, 18–20], line search methods [9, 22] and trust-region methods [26, 34, 37, 40].

Notation: Let us denote $\|\cdot\|$ as the Euclidean norm, the gradient of f by $g(x) := \nabla f(x)$ at the point x, $\mathbb{N}_0 := \mathbb{N} \cup \{0\}$ and $\|x\|_M := \sqrt{x^T M x}$ as the associated M-norm where $M \in \mathbb{R}^{n\times n}$ is a positive-definite matrix.

2010 *Mathematics Subject Classification*. Primary 90C30; Secondary 93E24, 34A34

Keywords. Unconstrained optimization, Conjugate gradient method, Nonmonotone line search method, Global convergence, Compressive sensing.

Received: 08 June 2016; Revised: 22 December 2016; Accepted: 01 March 2018

Communicated by Predrag Stanimirović

Email addresses: esmaeili47@yahoo.com (Hamid Esmaeili), m.rostami@basu.ac.ir (Majid Rostami), kimiaeim83@univie.ac.at (Morteza Kimiaei)

Conjugate gradient methods [13, 16, 22, 24, 30] are the powerful classes of iterative methods to solve unconstrained optimization problems, which are suitable especially for the large-scale problems due to the simplicity of their iterates and low memory requirements. Furthermore, these methods have strong local and global convergence properties [3, 11, 23, 31]. A nonlinear conjugate gradient method generates a sequence $\{x_k\}_{k\geq 0}$ by starting from an initial point $x_0 \in \mathbb{R}^n$ and using the iterative scheme

$$x_{k+1} := x_k + \alpha_k d_k, \quad k = 0, 1, \ldots, \tag{2}$$

where x_k is the current iterate, d_k is a search direction and $\alpha_k > 0$ is a step-size, determined by the exact or inexact line search methods. The exact line search methods are difficult or even impossible to seek in practical computation. Some inexact line search methods have been provided in [28] such as Armijo and Wolfe line searches. The Armijo condition is presented as monotone and nonmonotone strategies. The monotone Armijo line search guarantees the descent property of the sequence $\{f_k\}_{k\geq 0}$, i.e.,

$$f_{k+1} < f_k, \quad \forall k \geq 1, \tag{3}$$

in which $f_k := f(x_k)$. Although the nonmonotone strategies do not guarantee the descent property (3) are effective or even powerful in some iterates, especially when the iterates are trapped in a narrow curved valley of objective functions. Recently, several nonmonotone line search methods are presented that may decrease the number of function evaluations [21, 39].

In conjugate gradient methods, the direction d_k is obtained by using the gradient vector of new iterate and the previous direction. In other words, the direction d_k is defined by

$$d_k := \begin{cases} -g_k, & \text{if} \quad k = 0, \\ -g_k + \beta_k d_{k-1}, & \text{if} \quad k \geq 1, \end{cases} \tag{4}$$

where $g_k := g(x_k)$ and $\beta_k \in \mathbb{R}$ is a parameter that its different choices lead to produce various nonlinear conjugate gradient methods. The most well-known nonlinear conjugate gradient methods are such as

$$\beta_k^{FR} := \frac{\|g_k\|^2}{\|g_{k-1}\|^2}, \qquad \text{Fletcher-Reeves (FR) [16]} \tag{5}$$

$$\beta_k^{HS} := \frac{g_k^T y_{k-1}}{d_{k-1}^T y_{k-1}}, \qquad \text{Hestenes-Stiefel (HS) [24]} \tag{6}$$

$$\beta_k^{PR} := \frac{g_k^T y_{k-1}}{\|g_{k-1}\|^2}, \qquad \text{Polak-Ribière (PR) [31]} \tag{7}$$

$$\beta_k^{DY} := \frac{\|g_k\|^2}{d_{k-1}^T y_{k-1}}, \qquad \text{Dai-Yuan (DY) [13]} \tag{8}$$

$$\beta_k^{HZ} := \beta_k^{HS} - 2\|y_{k-1}\|^2 \frac{d_{k-1}^T g_k}{(d_{k-1}^T y_{k-1})^2}, \qquad \text{Hager-Zhang (HZ) [22]} \tag{9}$$

where $y_{k-1} := g_k - g_{k-1}$. In the case that the objective function $f(x)$ is strictly convex quadratic and the exact line search is used, all choices for the parameter β_k are equivalent, but each choice for this parameter leads to different performance for a general function cf. [28].

Content. In this paper, we produce a new spectral conjugate gradient method to solve unconstrained optimization problems while the β_k's of new approach will be produced based on β_k's of DY method. We use a kind of nonmonotone line search to determine the step-size in each iterate. The descent property and global convergence of proposed algorithm will be established. Numerical results show that our method has low computational cost. Furthermore, the new algorithm will be employed to solve compressive sensing problems.

The remaining of this paper is organized as follows. In Section 2, we describe a new spectral conjugate gradient algorithm to solve unconstrained optimization problems. In next section, descent property and global convergence of new algorithm will be investigated. Preliminary numerical results are reported in Section 4 for unconstrained optimization problems. In Section 5, we employ new approach to solve compressive sensing problems. Finally, some conclusions are given in Section 6.

2. New spectral conjugate gradient method

In this section, we first introduce a new parameter of our conjugate gradient method based on DY nonlinear conjugate gradient method. Next, the spectral parameter to guarantee the descent property of our algorithm will be presented and then described a New Spectral Conjugate Gradient Algorithm (NSCG) to solve (1).

In 1999, Dai and Yuan [13] presented a nonlinear conjugate gradient method that always generates descent direction by using the standard Wolfe conditions and its global convergence is obtained whenever the Lipschitz assumption holds for g. In NSCG Algorithm, wherever $g_k^T d_{k-1} > 0$, the new parameter β_k^N is introduced by using β_k^{DY}. In parameter β_k^N, the numerator is an efficiency convex combination from the maximum gradient norm value in some preceding iterates and a current gradient norm value. In other words, β_k^N is presented as follows:

$$\beta_k^N := \frac{\Gamma_k}{d_{k-1}^T y_{k-1}}, \tag{10}$$

where Γ_k is the convex combination

$$\Gamma_k := \eta\, g_{l_1(k)} + (1-\eta)\|g_k\|^2, \tag{11}$$

while $\eta \in (0,1)$ and

$$g_{l_1(k)} := \max_{0\le j\le n_1(k)} \|g_{k-j}\|^2, \;\; k \in \mathbb{N}_0, \tag{12}$$

in which $n_1(0) := 0$ and $0 \le n_1(k) \le \min\{n_1(k-1)+1, N_1\}$, with $N_1 \ge 0$. Note that $l_1(k)$ denotes one of the iterations where the maximum is attained. For the case that $g_k^T d_{k-1} \le 0$, we use β_k^{FR} as the conjugate gradient parameter in NSCG Algorithm.

Birgin and Martínez [6] were the first researches to introduce the spectral conjugate gradient method to solve unconstrained optimization problems. In spectral conjugate gradient methods, the direction is determined by

$$d_k := \begin{cases} -g_k, & \text{if} \quad k = 0, \\ -\theta_k g_k + \beta_k d_{k-1}, & \text{if} \quad k \ge 1, \end{cases} \tag{13}$$

in which θ_k is called as spectral parameter. To establish the descent property of NSCG Algorithm (see Lemma 3.2) and to improve the performance numerical, the spectral parameter is chosen by

$$\theta_k^1 := (1+\eta)\frac{g_{l_1(k)}}{\|g_k\|^2}, \tag{14}$$

whenever $g_k^T d_{k-1} > 0$. Otherwise, it is assumed that

$$\theta_k^2 := 1 + \frac{g_k^T d_{k-1}}{\|g_{k-1}\|^2}. \tag{15}$$

Finally, we use a nonmonotone Armijo-type condition to obtain the step-size α_k in order to complete our algorithm. Recently, Ahookhosh et al. [2] presented the following nonmonotone strategy:

$$f(x_k + \alpha_k d_k) \le R_k + \gamma\, \alpha_k g_k^T d_k, \tag{16}$$

for which $\gamma > 0$ and

$$R_k := \nu_k f_{l_2(k)} + (1 - \nu_k) f_k, \tag{17}$$

where $\nu_k \in [\nu_{\min}, \nu_{\max}]$, $\nu_{\min} \in [0, 1)$, $\nu_{\max} \in [\nu_{\min}, 1]$ and

$$f_{l_2(k)} := \max_{0 \le j \le n_2(k)} \{f_{k-j}\}, \quad k \in \mathbb{N}_0, \tag{18}$$

in which $n_2(0) := 0$ and $0 \le n_2(k) \le \min\{n_2(k-1) + 1, N_2\}$, with $N_2 \ge 0$. Note that $l_2(k)$ denotes one of the iterations where the maximum is attained.

Based on the above expression, a new iterate of NSCG Algorithm can be obtained by (2) for which d_k is computed by

$$d_k := \begin{cases} -\theta_k^1 g_k + \beta_k^N d_{k-1}, & \text{if} \quad g_k^T d_{k-1} > 0, \\ -\theta_k^2 g_k + \beta_k^{FR} d_{k-1}, & \text{if} \quad g_k^T d_{k-1} \le 0. \end{cases} \tag{19}$$

We can outline our new procedure in Algorithm 1.

Algorithm 1: A New Spectral Conjugate Gradient Algorithm (NSCG)

```
   Input: An initial point $x_0 \in \mathbb{R}^n$; $\eta, \sigma \in (0,1)$; $\gamma, \epsilon > 0$; $\nu_0 \in [\nu_{\min}, \nu_{\max}]$; $N_1, N_2 > 0$, $0 < \nu_{\min} < \nu_{\max} < 1$
          and $k_{\max}$.
1  begin
2      $n_1(0) = 0$; $n_2(0) = 0$; $f_0 := f(x_0)$; $g_0 := g(x_0)$; $d_0 := -g_0$; $R_0 := f_0$; $k := 0$;
3      while $\|g_k\| \ge \epsilon$ & $k \le k_{\max}$ do
4          $\alpha_k := 1$;
5          calculate $f(x_k + \alpha_k d_k)$;
6          while $f(x_k + \alpha_k d_k) > R_k + \gamma\, \alpha_k g_k^T d_k$ do
7              $\alpha_k := \sigma \alpha_k$;
8              compute $f(x_k + \alpha_k d_k)$;
9          end
10         $x_{k+1} = x_k + \alpha_k d_k$; $f_{k+1} := f(x_{k+1})$; $g_{k+1} := g(x_{k+1})$; $y_k := g_{k+1} - g_k$;
11         if $g_{k+1}^T d_k > 0$ then
12             update $n_1(k+1) = \min\{n_1(k), N_1\}$;
13             calculate $\Gamma_{k+1}$ by (11), $\beta_{k+1}^N$ by (10) and $\theta_{k+1}^1$ by (14);
14             $d_{k+1} := -\theta_{k+1}^1 g_{k+1} + \beta_{k+1}^N d_k$;
15         else
16             determine the parameter $\beta_{k+1}^{FR}$ by (5) and $\theta_{k+1}^2$ by (15);
17             $d_{k+1} := -\theta_{k+1}^2 g_{k+1} + \beta_{k+1}^{FR} d_k$;
18         end
19         update $n_2(k+1) = \min\{n_2(k), N_2\}$;
20         generate $R_{k+1}$ by (17);
21         $k \leftarrow k + 1$;
22     end
23     $x_b := x_k$;  $f_b := f_k$;
24 end
```

In this algorithm, the cycle starting from Line 6 to Line 9 is called **backtraking loop**. Such an algorithm has many benefits some of which are listed as follows:

- Spectral conjugate gradient algorithms have low requirement memory and strong local and global convergence properties, especially for large-scale problems, see [3, 11, 23, 31].

- It is believed that we need to produce a larger (smaller) parameter β_k whenever iterations are far away from (near) the optimizer [23, 28] but the DY method can not perform this fact; because the numerator of β_k^{DY} in the cases where iterations are near the optimizer may tends to zero. On the other hand, since the numerator of this patameter is the quantity $\|g_k\|^2$ with high oscillation, it is possible to be small far away from the optimizer, leading to the unsuitable numerical reasults. To overcome these shortcomings, NSCG Algorithm by using advantages of $g_{l_1(k)}$, concidered as (12), contorols the amount β_k^N. Let us now describe how to this process. This fact that $g_{l_1(k)}$ is maximum gradient norm value in some preceding iterations will prevent producing the very smaller β_k^N whenevere iterations are not near the optimizer; in addition, this fact along with how to update $n_1(k)$ (Line 12) prevents producing the large β_k^N near the optimizer.

3. Convergence theory

We first investigate the descent property and then global convergence of Algorithm 1 in this section. Consider the following assumptions to analyze the convergence results of the proposed algorithm:

(H1) For any $x_0 \in \mathbb{R}^n$, the level set $L(x_0) := \{x \in \mathbb{R}^n | f(x) \le f(x_0)\}$ is bounded.

(H2) The gradient of f is Lipschitz continuous, i.e., there exists constant $L > 0$ such that

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

To simplify the notation of global convergence, let us define

$$\mathcal{I}_1 := \{k|\ g_k^T d_{k-1} > 0\}, \quad \mathcal{I}_2 := \{k|\ g_k^T d_{k-1} \le 0\},$$

where $\mathcal{I}_1$ contains the iterations of the extended DY method with the spectral parameter θ_k^1 and $\mathcal{I}_2$ contains the iterations of FR method with the spectral parameter θ_k^2.

Lemma 3.1. *Let $\{x_k\}_{k\ge0}$ be the sequence generated by Algorithm 1. Then, for any $k \in \mathcal{I}_1$, we have $\|g_k\|^2 \le \Gamma_k \le g_{l_1(k)}$.*

Proof. Using (11) and this fact that $\|g_k\|^2 \le g_{l_1(k)}$, we get

$$\|g_k\|^2 = \eta\|g_k\|^2 + (1-\eta)\|g_k\|^2 \le \eta g_{l_1(k)} + (1-\eta)\|g_k\|^2 = \Gamma_k \le \eta g_{l_1(k)} + (1-\eta) g_{l_1(k)} = g_{l_1(k)}.$$

□

The following lemma ensures the descent property of the NSCG Algorithm, for any $k \in \mathcal{I}_1 \cup \mathcal{I}_2$.

Lemma 3.2. *Suppose that d_k is the generated direction by Algorithm 1. Then, d_k is a descent direction, i.e., $g_k^T d_k < 0$.*

Proof. We use the induction over k to prove this lemma. First, for $k = 0$, it can simply be obtained by $g_k^T d_k = -\|g_k\|^2 < 0$. By using the induction hypothesis, we have $g_{k-1}^T d_{k-1} < 0$. If $k \in \mathcal{I}_1$, then

$$d_{k-1}^T y_{k-1} = g_k^T d_{k-1} - g_{k-1}^T d_{k-1} > g_k^T d_{k-1} > 0,$$

leading to

$$0 < \frac{g_k^T d_{k-1}}{d_{k-1}^T y_{k-1}} < 1. \tag{20}$$

Recalling (10) and (19), along with Lemma 3.1 give

$$\begin{aligned} g_k^T d_k &= -\theta_k^1 \|g_k\|^2 + \frac{\Gamma_k}{d_{k-1}^T y_{k-1}} g_k^T d_{k-1} \\ &\le -\theta_k^1 \|g_k\|^2 + \frac{g_{l_1(k)}}{d_{k-1}^T y_{k-1}} g_k^T d_{k-1}. \end{aligned}$$

From (14) and (20), we get

$$g_k^T d_k < -\theta_k^1 \|g_k\|^2 + g_{l_1(k)} = -(1+\eta)\frac{g_{l_1(k)}}{\|g_k\|^2}\|g_k\|^2 + g_{l_1(k)} = -\eta g_{l_1(k)} < 0.$$

This inequality shows that d_k is a descent direction.
For $k \in \mathcal{I}_2$, (15) and (19) result in

$$\begin{aligned} g_k^T d_k &= -\theta_k^2 \|g_k\|^2 + \beta_k^{FR} g_k^T d_{k-1} \\ &= -\Big(1 + \frac{g_k^T d_{k-1}}{\|g_{k-1}\|^2}\Big)\|g_k\|^2 + \frac{\|g_k\|^2}{\|g_{k-1}\|^2} g_k^T d_{k-1} \\ &= -\|g_k\|^2 < 0. \end{aligned} \tag{21}$$

Hence, d_k satisfies the descent property, for any $k \in \mathcal{I}_1 \cup \mathcal{I}_2$. □

Lemma 3.3. *Suppose that the sequence $\{x_k\}_{k\geq 0}$ is generated by Algorithm 1 and (H1) holds. Then, we have*
(1) the sequence $\{f_{l_2(k)}\}_{k\geq 0}$ is non-increasing,
(2) $\lim_{k\to\infty} f_{l_2(k)} = \lim_{k\to\infty} f_k$,
(3) $\lim_{k\to\infty} R_k = \lim_{k\to\infty} f_k$.

Proof. See Lemmas 2.1 and 3.2 in [2]. □

Lemma 3.4. *Suppose that d_k is a descent direction and (H1) and (H2) hold. Then*

$$\sum_{l_2(k)-1} \frac{\Big(g_{l_2(k)-1}^T d_{l_2(k)-1}\Big)^2}{\|d_{l_2(k)-1}\|^2} < \infty.$$

Proof. First, the definition of $f_{l_2(k)}$ implies that

$$R_k = \nu_k f_{l_2(k)} + (1-\nu_k) f_k \leq \nu_k f_{l_2(k)} + (1-\nu_k) f_{l_2(k)} = f_{l_2(k)},$$

leading to

$$f(x_k + \alpha_k d_k) \leq f_{l_2(k)} + \gamma \alpha_k g_k^T d_k. \tag{22}$$

The mean-value theorem, Cauchy-Schwarz inequality and (H2) result in

$$\begin{aligned} f(x_k + \alpha_k d_k) - f_{l_2(k)} &\leq f(x_k + \alpha_k d_k) - f(x_k) \\ &= \alpha_k g(x_k + \alpha_k t_k d_k)^T d_k \\ &= \alpha_k g_k^T d_k + \alpha_k \Big(g(x_k + \alpha_k t_k d_k) - g_k\Big)^T d_k \\ &\leq \alpha_k g_k^T d_k + L\alpha_k^2 \|d_k\|^2, \end{aligned}$$

in which $t_k \in (0,1)$. As long as the condition (16) is not satisfied

$$\gamma \alpha_k g_k^T d_k < f(x_k + \alpha_k d_k) - f_k \leq \alpha_k g_k^T d_k + L\alpha_k^2 \|d_k\|^2,$$

we obtain

$$\alpha_k g_k^T d_k + L\alpha_k^2 \|d_k\|^2 - \gamma \alpha_k g_k^T d_k > 0,$$

$$(1-\gamma)\alpha_k g_k^T d_k + L\alpha_k^2 \|d_k\|^2 > 0,$$

yielding to

$$L\alpha_k^2\|d_k\|^2 > (\gamma - 1)\alpha_k g_k^T d_k,$$

therefore

$$\alpha_k > \frac{(\gamma - 1)g_k^T d_k}{L\|d_k\|^2}.$$

From (22), we now have

$$\sum_k (-\gamma\alpha_k g_k^T d_k) \leq \sum_k (f_{l_2(k)} - f(x_k + \alpha_k d_k)). \tag{23}$$

Since $l_2(k) \leq k$, replacing k by $l_2(k) - 1$ in (23) along with Lemma 3.3 implies that

$$\sum_{l_2(k)-1} (-\gamma\alpha_{l_2(k)-1} g_{l_2(k)-1}^T d_{l_2(k)-1}) \leq \sum_{l_2(k)-1} \left(f_{l_2(l_2(k)-1)} - f_{l_2(k)}\right) < C,$$

in which $C > 0$. Therefore,

$$C > \sum_{l_2(k)-1} (-\gamma\alpha_{l_2(k)-1} g_{l_2(k)-1}^T d_{l_2(k)-1}) > \sum_{l_2(k)-1} \frac{(1-\gamma)\gamma}{L} \frac{\left(g_{l_2(k)-1}^T d_{l_2(k)-1}\right)^2}{\|d_{l_2(k)-1}\|^2},$$

giving

$$\sum_{l_2(k)-1} \frac{\left(g_{l_2(k)-1}^T d_{l_2(k)-1}\right)^2}{\|d_{l_2(k)-1}\|^2} < \infty.$$

□

Lemma 3.5. *For any $k \in \mathcal{I}_1$, we have*

$$0 \leq \beta_k^N \leq \frac{g_k^T d_k}{g_{k-1}^T d_{k-1}}.$$

Proof. From (11), it is clear that $\Gamma_k \geq 0$. Also, we have $g_k^T d_{k-1} > 0$ for any $k \in \mathcal{I}_1$, therefore Lemma 3.2 implies

$$d_{k-1}^T y_{k-1} = g_k^T d_{k-1} - g_{k-1}^T d_{k-1} > 0.$$

By using (10), the parameter β_k^N is non-negative.

For $k \in \mathcal{I}_1$, from (10), we have

$$\eta g_{l_1(k)} + (1-\eta)\|g_k\|^2 = \beta_k^N d_{k-1}^T y_{k-1} = \beta_k^N g_k^T d_{k-1} - \beta_k^N g_{k-1}^T d_{k-1}. \tag{24}$$

By substituting the $\beta_k^N g_k^T d_{k-1} = g_k^T d_k + \theta_k^1 \|g_k\|^2$ in (24) and with the use of (14), we obtain

$$\beta_k^N = \frac{g_k^T d_k + g_{l_1(k)} + (\eta - 1)\|g_k\|^2}{g_{k-1}^T d_{k-1}}.$$

Since $\|g_k\|^2 \leq g_{l_1(k)}$, then $g_{l_1(k)} + (\eta - 1)\|g_k\|^2 \geq 0$ and therefore

$$\frac{g_{l_1(k)} + (\eta - 1)\|g_k\|^2}{g_{k-1}^T d_{k-1}} \leq 0.$$

Finally, we get

$$\beta_k^N = \frac{g_k^T d_k}{g_{k-1}^T d_{k-1}} + \frac{g_{l_1(k)} + (\eta - 1)\|g_k\|^2}{g_{k-1}^T d_{k-1}} \leq \frac{g_k^T d_k}{g_{k-1}^T d_{k-1}}.$$

□

In the next theorem, we prove global convergence of new algorithm.

Theorem 3.6. *Let d_k be a descent direction and $\{x_k\}_{k\geq 0}$ be the generated sequence by Algorithm 1. Moreover, suppose that (H1) and (H2) hold. Then* $\lim_{k\longrightarrow\infty} \inf \|g_k\| = 0$.

Proof. By contradiction, suppose that there exists a constant $\zeta > 0$ such that $\|g_k\| > \zeta$, for any k. Let d_k be the generated spectral direction by (19). For $i = 1, 2$ and $\beta_k = \beta_k^N, \beta_k^{FR}$, we can obtain

$$\begin{aligned}
\|d_k\|^2 &= (-\theta_k^i g_k + \beta_k d_{k-1})^T(-\theta_k^i g_k + \beta_k d_{k-1}) \\
&= \left(\theta_k^i \|g_k\|\right)^2 - 2\theta_k^i \beta_k d_{k-1}^T g_k + \beta_k^2 \|d_{k-1}\|^2 \\
&= \left(\theta_k^i \|g_k\|\right)^2 - 2\theta_k^i (d_k + \theta_k^i g_k)^T g_k + \beta_k^2 \|d_{k-1}\|^2 \\
&= \left(\theta_k^i \|g_k\|\right)^2 - 2\theta_k^i g_k^T d_k - 2\left(\theta_k^i \|g_k\|\right)^2 + \beta_k^2 \|d_{k-1}\|^2 \\
&= \beta_k^2 \|d_{k-1}\|^2 - 2\theta_k^i g_k^T d_k - \left(\theta_k^i \|g_k\|\right)^2.
\end{aligned}$$

By dividing both sides of this equality in $(g_k^T d_k)^2$, we have

$$\frac{\|d_k\|^2}{(g_k^T d_k)^2} = \frac{\beta_k^2 \|d_{k-1}\|^2}{(g_k^T d_k)^2} - 2\theta_k^i \frac{g_k^T d_k}{(g_k^T d_k)^2} - \frac{\left(\theta_k^i \|g_k\|\right)^2}{(g_k^T d_k)^2}. \tag{25}$$

The remaindering of proof follows in the two cases:

Case (i): For $k \in \mathcal{I}_1$, by (25) and Lemma 3.5, we have

$$\frac{\|d_k\|^2}{(g_k^T d_k)^2} = \frac{\left(\beta_k^N\right)^2 \|d_{k-1}\|^2}{(g_k^T d_k)^2} - \Big(\frac{\theta_k^1 \|g_k\|}{g_k^T d_k} + \frac{1}{\|g_k\|}\Big)^2 + \frac{1}{\|g_k\|^2} \leq \frac{\|d_{k-1}\|^2}{(g_{k-1}^T d_{k-1})^2} + \frac{1}{\|g_k\|^2}.$$

Case (ii): For $k \in \mathcal{I}_2$, from (21) we have $g_k^T d_k = -\|g_k\|^2$. By setting this equation and (5) in (25) imply that

$$\begin{aligned}
\frac{\|d_k\|^2}{(g_k^T d_k)^2} = \frac{\|d_k\|^2}{\|g_k\|^4} &= \frac{\left(\beta_k^{FR}\right)^2 \|d_{k-1}\|^2}{\|g_k\|^4} + \frac{2\theta_k^2}{\|g_k\|^2} - \frac{\left(\theta_k^2 \|g_k\|\right)^2}{\|g_k\|^4} \\
&= \frac{\|d_{k-1}\|^2}{\|g_{k-1}\|^4} - \Big(\frac{1}{\|g_k\|} - \frac{\theta_k^2}{\|g_k\|}\Big)^2 + \frac{1}{\|g_k\|^2} \\
&\leq \frac{\|d_{k-1}\|^2}{(g_{k-1}^T d_{k-1})^2} + \frac{1}{\|g_k\|^2}.
\end{aligned}$$

For both cases, since $\|g_k\| > \zeta$, we have

$$\frac{\|d_k\|^2}{(g_k^T d_k)^2} \leq \frac{\|d_{k-1}\|^2}{(g_{k-1}^T d_{k-1})^2} + \frac{1}{\|g_k\|^2} \leq \frac{\|d_{k-2}\|^2}{(g_{k-2}^T d_{k-2})^2} + \frac{1}{\|g_{k-1}\|^2} + \frac{1}{\|g_k\|^2} \leq \cdots \leq \sum_{i=0}^{k} \frac{1}{\|g_i\|^2} \leq \frac{k}{\zeta^2}.$$

Therefore,

$$\frac{(g_k^T d_k)^2}{\|d_k\|^2} \geq \frac{\zeta^2}{k}. \tag{26}$$

Without loss of generality, we can let $k := l_2(k) - 1$ in (26). Hence, we get

$$\sum_{l_2(k)-1} \frac{\left(g_{l_2(k)-1}^T d_{l_2(k)-1}\right)^2}{\|d_{l_2(k)-1}\|^2} \geq \sum_{l_2(k)-1} \left(\frac{\zeta^2}{l_2(k)-1}\right) = \infty, \tag{27}$$

which contradicts with Lemma 3.4. Therefore, the desired result is valid. □

4. Preliminary numerical experiments

In this section, we investigate the efficiency of NSCG Algorithm to solve unconstrained optimization problems. In experiments, we compare NSCG Algorithm with several versions of conjugate gradient methods, the details of which are as follows:

- CGDY: conjugate gradient proposed by Dai–Yuan [13],
- CGFR: conjugate gradient proposed by Fletcher–Reeves [16],
- CGPR: conjugate gradient proposed by Polak–Ribière [31],
- CGHZ: conjugate gradient proposed by Hager–Zhang [22].

All algorithms are implemented in Matlab 2011 programming environment on a 2.3Hz Intel core i3 processor laptop and 4GB of RAM with the double precision data type in Linux operations system. Our experiments are performed on a large-scale test functions of the nonlinear unconstrained optimization problems from the CUTEst [10] library. The initial points are standard ones proposed in CUTEst. Since we are interested in large-scale problems, we only consider problems with dimension at least 1000. Given in Table 1 are these problems.

Here, we choose the parameters $\sigma := 0.5$, $\gamma := 10^{-4}$ and $N_1 = N_2 := 10$. The parameter ν_k is updated by

$$\nu_k := \begin{cases} \nu_0/2, & \text{if } k = 1, \\ (\nu_{k-1} + \nu_{k-2})/2, & \text{if } k > 1, \end{cases}$$

in which $\nu_0 := 0.15$.

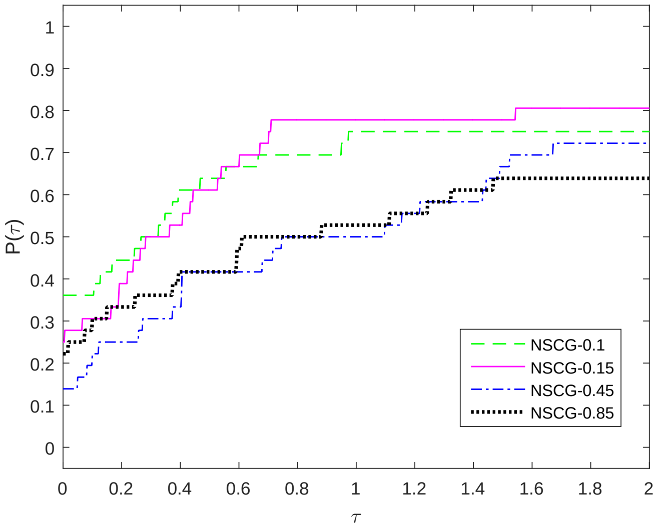
All algorithms are stopped whenever $\|g_k\| < 10^{-5}$ or the total number of iterates exceeds 20000. We have implemented the NSCG Algorithm for several values of $\eta \in \{0.1, 0.15, 0.45, 0.85\}$, considered as NSCG-0.1, NSCG-0.15, NSCG-0.45 and NSCG-0.85, respectively. We have used the performance profiles proposed by Dolan and Moré [14] to display the performance of each algorithm where N_i, N_f and C_t indicate the total number of iterations, the total number of function evaluations and CPU times, respectively. Let us consider P as designates the percentage of problems which are solved within a factor τ of the best solver. The horizontal axis of the figure gives the percentage of the test problems for which a method is the fastest (efficiency), while the vertical axis gives the percentage of the test problems that were successfully solved by each method (robustness). Here, we have illustrated the results of NSCG Algorithm for values of parameter

Table 1: List of test functions

Problem name	Dim	Problem name	Dim
ARWHEAD	3000	DQRTIC	5000
BRYBND	10000	EDENSCH	10000
CHAINWOO	1000	ENGVAL1	10000
CRAGGLVY	1000	FMINSRF2	1024
DIXMAANA	9000	MOREBV	15000
DIXMAANB	9000	NLMSURF	5625
DIXMAANC	9000	PENALTY1	5000
DIXMAAND	9000	POWELLSG	1000
DIXMAANE	9000	POWER	1000
DIXMAANF	9000	QUARTC	5000
DIXMAANG	3000	RAYBENDL	2050
DIXMAANH	3000	SCHMVETT	2000
DIXMAANI	9000	SPARSQUR	10000
DIXMAANJ	3000	SPMSRTLS	1000
DIXMAANK	9000	SROSENBR	5000
DIXMAANL	9000	TOINTGSS	5000
DIXON3DQ	1000	VARDIM	3000
DQDRTIC	5000	WOODS	10000

η in Figure 1. Subfigures (a)-(c) of Figure 1 show that NSCG-0.1 is better than NSCG-0.15, NSCG-0.45 and NSCG-0.85 about 37%, 33% and 30% of the most wins in N_i, N_f and C_t, respectively.

In addition, NSCG-0.1 is compared with other conjugate gradient methods in Figure 2. Subfigures (a)-(c) of Figure 2 show that NSCG-0.1 is better than CGDY, CGFR, CGPR and CGHZ about 30%, 39% and 30% of the most wins in N_i, N_f and C_t, respectively. Summarizing the results of this section, we see that NSCG-0.1 is comparable for large-scale unconstrained optimization problems.



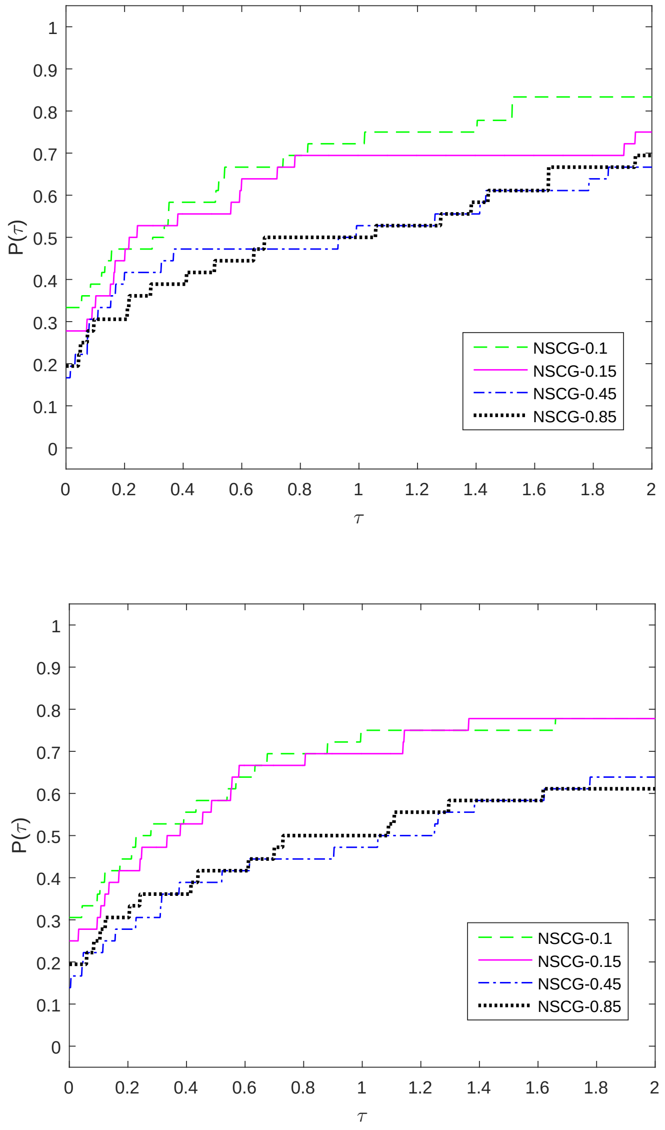


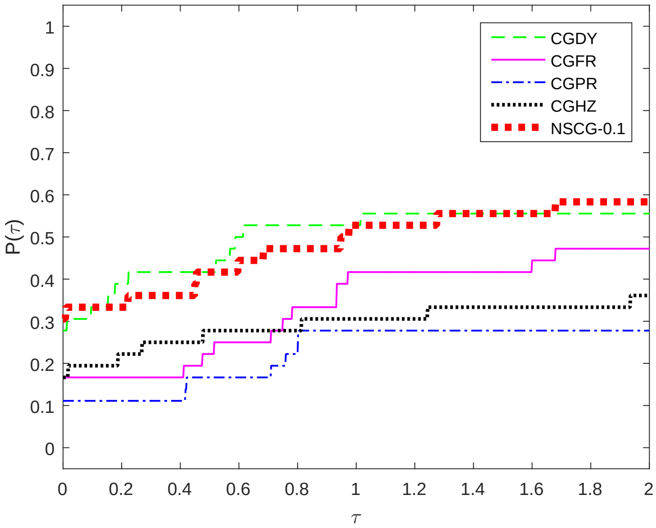
Figure 1: A comparison among NSCG-0.1, NSCG-0.15, NSCG-0.45 and NSCG-0.85 by performance profiles using the measures N_i, N_f and C_t: (a) displays the number of iterations (top); (b) shows the number of function evaluations (middle); (c) displays CPU times (down), respectively.

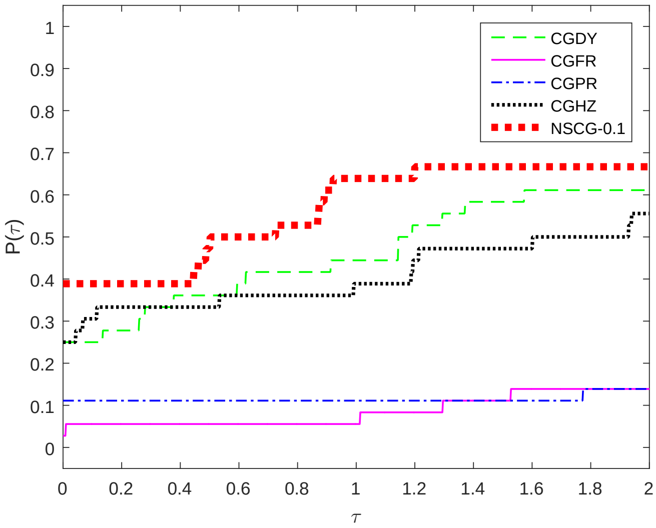
5. Compressive sensing

Compressive sensing (CS) is a framework for signal reconstruction from a measurement vector, which is inherently sparse. A signal $x \in \mathbb{R}^n$ is called sparse if most of its components are zero, i.e., $\|x\|_0 := |\{i \,|\, x_i \neq 0\}|$ is small or there exists an orthonormal basis Φ such that $x := \Phi c$ with c being sparse. Note that Φ is the matrix with orthonormal column vectors. CS plays an important role in medical and astronomical imaging, file restoration, image and video coding and other applications. This new research field was presented by Donoho [15] and Candès et al. [7].

Put $A \in \mathbb{R}^{m\times n}$, $m \ll n$, is called sensing matrix. Then, CS problem is recover sparse signal x from

$$Ax = b,$$





where $b \in \mathbb{R}^m$ is the vector of observations. The CS problem can be expressed as a constrained optimization problem

$$\begin{array}{ll} \min & \|x\|_0 \\ \text{s.t.} & Ax = b \\ & x \in \mathbb{R}^n. \end{array} \tag{28}$$

Therefore, we search for the most sparse vector consistent with the measured data $Ax = b$. Unfortunately, ℓ_0-minimization is NP-hard in general. Hence, Chen et al. in [8] replaced the ℓ_0-norm by the closest convex norm, which is the ℓ_1-norm. This leads to the minimization problem

$$\begin{array}{ll} \min & \|x\|_1 \\ \text{s.t.} & Ax = b \\ & x \in \mathbb{R}^n. \end{array} \tag{29}$$

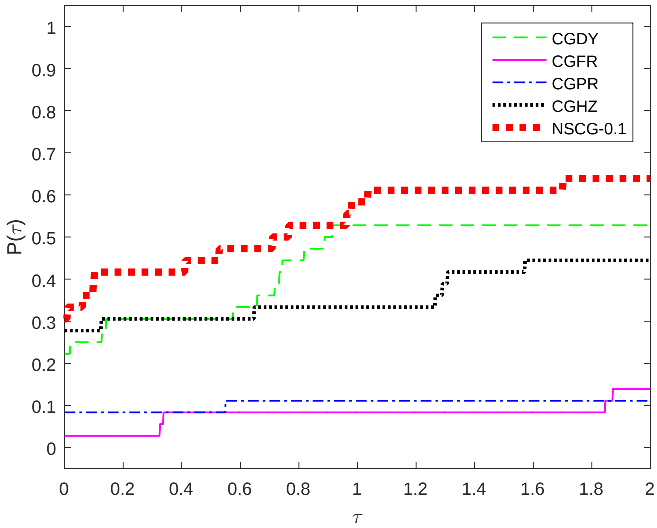
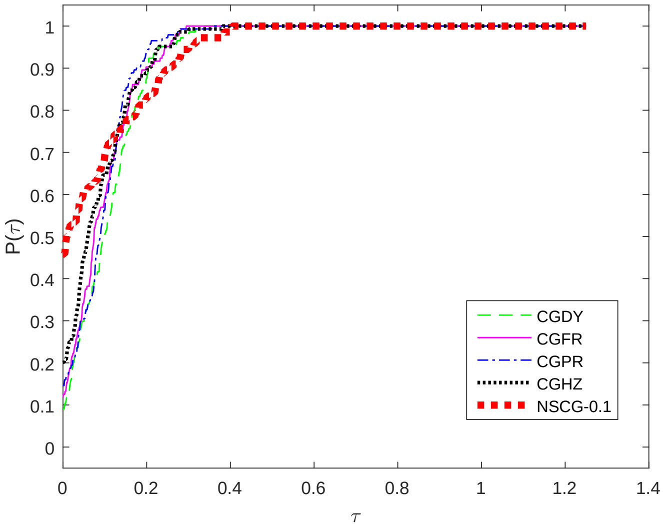


Figure 2: A comparison among CGDY, CGFR, CGPR, CGHZ and NSCG-0.1 by performance profiles using the measures N_i, N_f and C_t: (a) displays the number of iterations (top); (b) shows the number of function evaluations (middle); (c) displays CPU times (down), respectively.

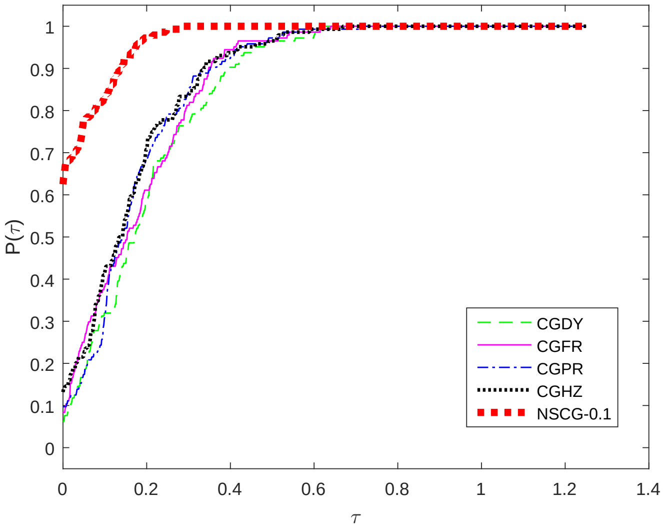


If the measurements are affected by noise, a conic constraint is required; i.e., the minimization problem needs to be changed to

$$\begin{array}{ll} \min & \|x\|_1 \\ \text{s.t.} & \|Ax-b\|_M^2 \le \xi \\ & x \in \mathbb{R}^n, \end{array} \tag{30}$$

with a carefully chosen $\xi > 0$. For a regularization parameter $\mu > 0$, the problem (30) is equivalent to unconstrained optimization problem

$$\min \Psi(x) := f(x) + \mu\|x\|_1, \tag{31}$$



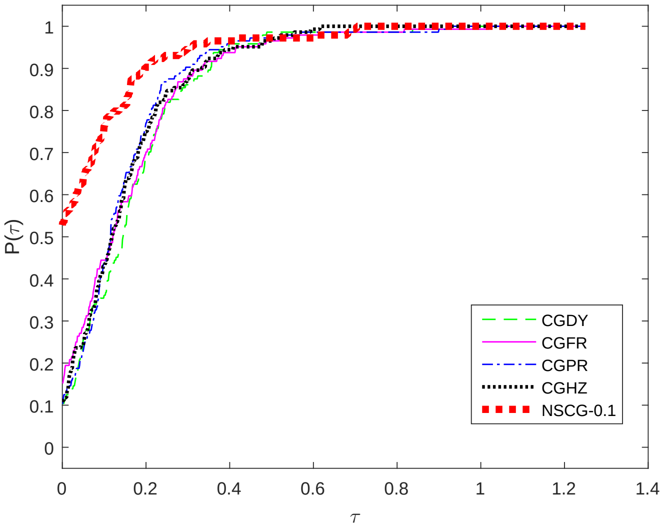


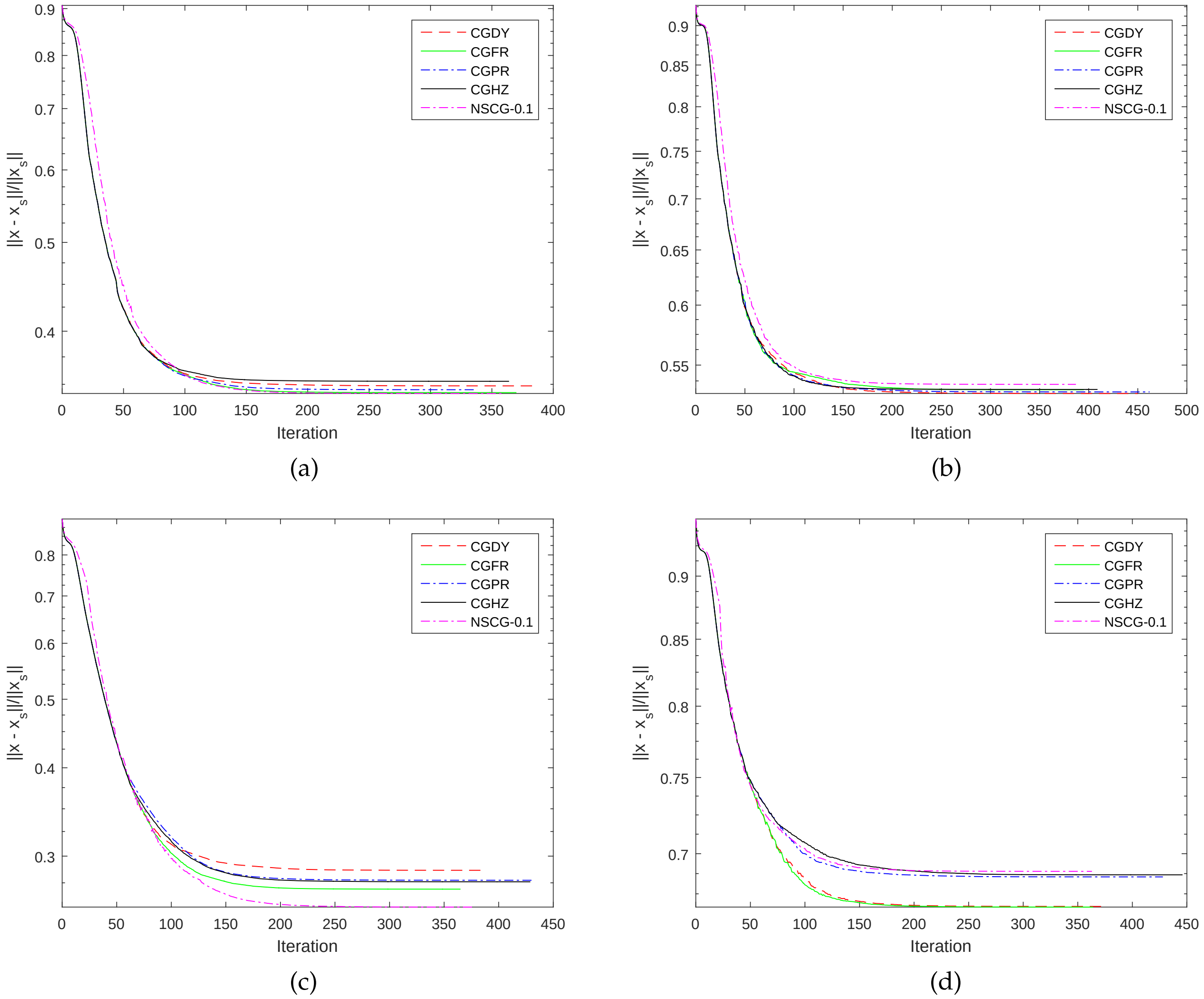
Figure 3: A comparison among CGDY, CGFR, CGPR, CGHZ and NSCG-0.1 by performance profiles using the measures N_i, N_f and C_t: (a) displays the number of iterations (top); (b) shows the number of function evaluations (middle); (c) displays CPU times (down), respectively.

in which $f(x) := \frac{1}{2}\|Ax - b\|_M^2$. Here, M is the $m \times m$ identity matrix. Since $\|x\|_1$ is not differentiable, (31) is not smooth; hence, the smooth function is used

$$F(x) := \sum_{i=1}^{n} \sqrt{x_i^2 + \varepsilon}, \tag{32}$$

in which $\varepsilon > 0$, to approximate the non-smooth term $\|x\|_1$ in problem (31). It is clear that $F(x)$ tends to $\|x\|_1$ whenever $\varepsilon \to 0$. So, we have the smooth problem

$$\min \widetilde{\Psi}(x) := f(x) + \mu \sum_{i=1}^{n} \sqrt{x_i^2 + \varepsilon}, \tag{33}$$



(a)

(b)

(c)

(d)

Figure 4: A comparison among all algorithms. (a)-(d) show the diagrams of relative errors versus iterations for the **Type 1** of matrix A, respectively, with the noise levels (a) $\sigma_1 = \sigma_2 = 10^{-1}$, $\rho = 0.1$, $\delta = 0.3$; (b) $\sigma_1 = \sigma_2 = 10^{-1}$, $\rho = 0.2$, $\delta = 0.2$; (c) $\sigma_1 = \sigma_2 = 10^{-7}$, $\rho = 0.2$, $\delta = 0.3$; (d) $\sigma_1 = \sigma_2 = 10^{-7}$, $\rho = 0.3$, $\delta = 0.2$.

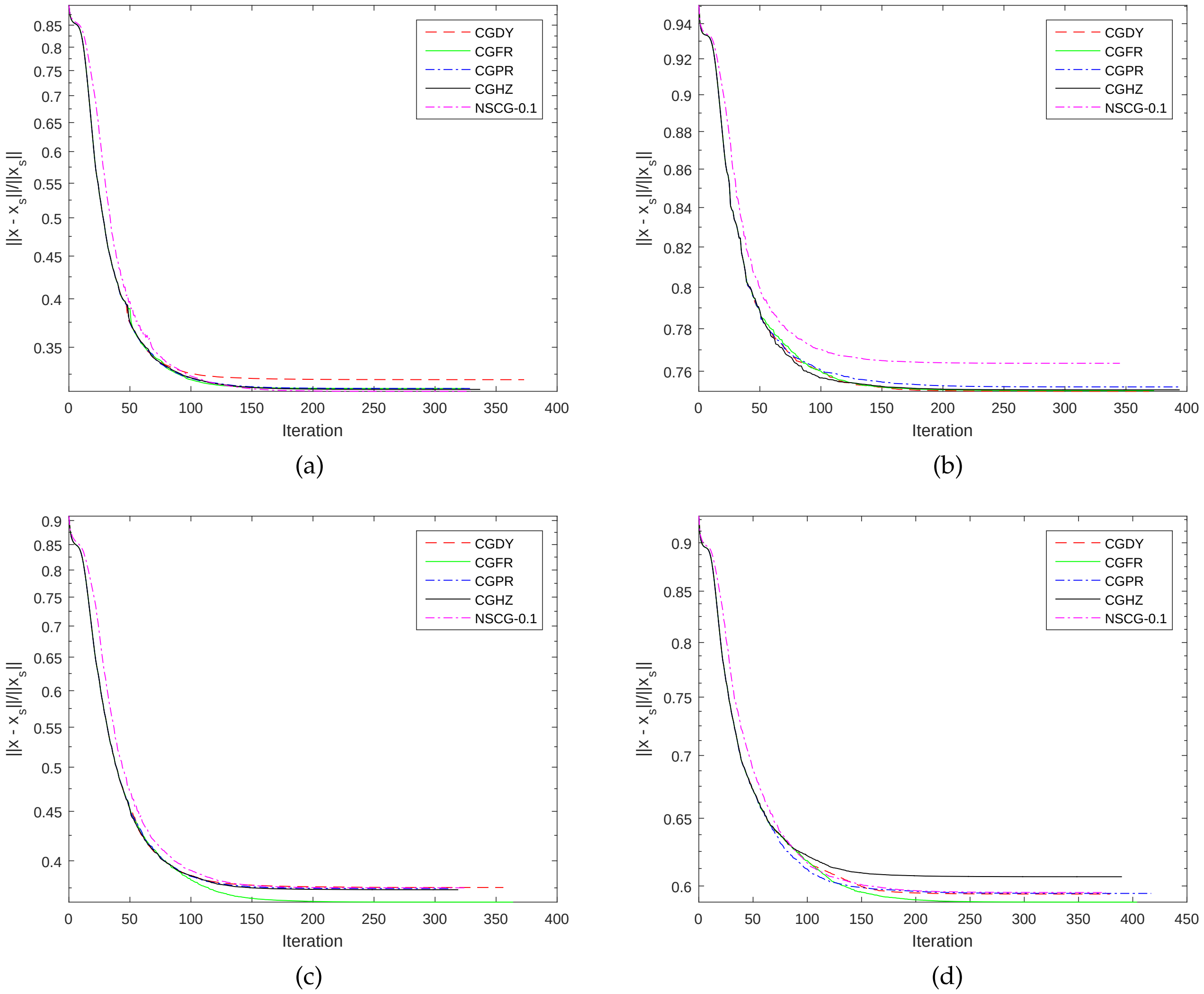
for which $\mu := 2^{-4}$ and the gradient of $\widetilde{\Psi}(x)$ is considered as $\nabla\widetilde{\Psi}(x) := \nabla f(x) + \mu\nabla F(x)$, in which

$$\nabla f(x) := A^T M(Ax - b), \quad \nabla F(x) := \Big(\frac{x_1}{\sqrt{x_1^2 + \varepsilon}}, \cdots, \frac{x_n}{\sqrt{x_n^2 + \varepsilon}}\Big).$$

Now, we use NSCG-0.1 Algorithm to solve smooth problem (33) and compare it with several versions of conjugate gradient methods, presented in Section 4, for which d_k is computed by

$$d_k := \begin{cases} -\theta_k^1 \nabla\widetilde{\Psi}(x_k) + \vartheta\beta_k^N d_{k-1}, & \text{if} \quad \nabla\widetilde{\Psi}(x_k)^T d_{k-1} > 0, \\ -\theta_k^2 \nabla\widetilde{\Psi}(x_k) + \vartheta\beta_k^{FR} d_{k-1}, & \text{if} \quad \nabla\widetilde{\Psi}(x_k)^T d_{k-1} \le 0, \end{cases}$$

where $\vartheta = 10^{-9}$, θ_k^1 and θ_k^2 compute by (14) and (15), respectively. All parameters of proposed algorithms are choosen similar to the previous section. We consider an average of five implementations of these algorithms.



(a)

(b)

(c)

(d)

Figure 5: A comparison among all algorithms. (a)-(d) show the diagrams of relative errors versus iterations for the **Type 2** of matrix A, respectively, with the noise levels (a) $\sigma_1 = \sigma_2 = 10^{-1}$, $\rho = 0.1$, $\delta = 0.3$; (b) $\sigma_1 = \sigma_2 = 10^{-1}$, $\rho = 0.2$, $\delta = 0.2$; (c) $\sigma_1 = \sigma_2 = 10^{-7}$, $\rho = 0.2$, $\delta = 0.3$; (d) $\sigma_1 = \sigma_2 = 10^{-7}$, $\rho = 0.3$, $\delta = 0.2$.

To solve (33), all algorithms employ the parameters $m := \delta n$ (the number of observations) and $o := \rho m$ (the number of nonzeros) in which $n \in \{2^{10}, 2^{11}, 2^{12}\}$ and the amounts δ and ρ are given in Table 2.

We now introduce how to contaminate $\bar{x}$ and b by impulse noise in the following in the procedure 1. In this procedure, Line 2 returns the n-by-1 matrix containing zeros, Line 3 returns a random permutation of the integers from 1 to n, Line 4 returns an o-by-1 matrix containing pseudo random values drawn from the standard normal distribution and Lines 5 and 7 return $\bar{x}$ and b, respectively, contaminated by impulsive noise. In addition, σ_1 and σ_2 are presented in Table 3.

By the cosidered amounts for σ_1, σ_2, δ, ρ and the four proposed matrices for A, as follows:

Type 1. Gaussian matrix whose elements are generated from i.i.d normal distribution $\mathcal{N}(0, 1)$ (randn(m, n)),
Type 2. Gaussian matrix whose elements are generated from i.i.d normal distribution $\mathcal{N}(0, 1)$, which has their columns scaled to unit norm,

```
Procedure 1(for generating $\bar{x}$ and $b$)
Input: $n, o, \sigma_1, \sigma_2$ and $A$
begin
    $\bar{x} := zeros(n, 1);$
    $p := \mathrm{randperm}(n);$
    $\bar{x}(p(1 : o)) := 2\mathrm{randn}(o, 1);$
    $\bar{x} := \bar{x} + \sigma_2\mathrm{randn}(n, 1);$
    $s_I := \mathrm{randn}(m, 1);$
    $b := A\bar{x} + \sigma_1 s_I;$
end
Output: $\bar{x}$ and $b$
```

Type 3. Hadamard matrix H which is a matrix of 1′s and -1′s whose columns are orthogonal,
Type 4. Partial 1-d discrete cosine transform matrix,
we get a total of 144 problems.

Table 2: Amounts of parameters ρ and δ

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

Table 3: Amounts of parameters σ_1 and σ_2

σ_1	σ_2	σ_1	σ_2
10^{-1}	10^{-1}	10^{-5}	10^{-5}
10^{-3}	10^{-3}	10^{-7}	10^{-7}

We update ε_k for function $F(x)$ in each iteration by the following formula

$$\varepsilon_k := \max\{0.5\varepsilon_{k-1}, \epsilon_m\},$$

where $\varepsilon_0 := 50$ and ϵ_m is precision machine.

In Subfigures (a)-(c) of Figure 3, NSCG-0.1 is compared with CGDY, CGFR, CGPR and CGHZ to solve smooth CS problem. From Figure 3, it is clear that NSCG-0.1 in N_i, N_f and C_t wins about 46%, 62% and 53%, respectively. Finally, Figures 4 and 5 report that NSCG-0.1 outperforms others algorithms with respect to the relative error of step points via iterations.

6. Conclusion

In this paper, we have proposed a new spectral conjugate gradient method to solve unconstrained optimization and CS problems. In the sense, the new conjugate gradient parameter has been introduced based on DY method that try to produce the larger step-size far away from the optimizer and the smaller step-size close to it. Also, the spectral parameter guarantees the descent property of the new approach. Moreover, the global convergence of the new algorithm is investigated under some standard assumptions. Finally, numerical experiments on the set of large-scale standard test problems point out that the proposed algorithm is efficient to solve unconstrained optimization problems. Furthermore, the new algorithm is more efficient than some versions of conjugate gradient methods to solve CS problem.

7. Acknowledgements

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] M. Ahookhosh, K. Amini, H. Nosratipour, An inexact line search approach using modified nonmonotone strategy for unconstrained optimization, Numerical Algorithms 66 (2014) 49–78.

[2] M. Ahookhosh, K. Amini, M. R. Peyghami, A nonmonotone trust-region line search method for large-scale unconstrained optimization, Applied Mathematical Modelling 36 (2012) no. 1, 478–487.

[3] M. Al-Baali, Descent property and global convergence of the Fletcher-Reeves method with inexact line search, IMA J. Numer. Anal. 5 (1985) 121–124.

[4] N. Andrei, A Dai-Yuan conjugate gradient algorithm with sufficient descent and conjugacy conditions for unconstrained optimization, Applied Mathematics Letters 21 (2008) 165–171.

[5] J. Barzilai, J. M. Borwein, Two point step size gradient method, IMA J. Numer. Anal. 8 (1988) 141–148.

[6] E. G. Birgin, J. M. Martínez, A spectral conjugate gradient method for unconstrained optimization, Applied Mathematics and Optimization 43 (2001) no. 2, 117–128.

[7] E. Candès, J. Romberg, T. Tao, Stable signal recovery from incomplete and inaccurate measurements, Comm. Pure Appl. Math. 59 (2006) 1207–1233.

[8] S. S. Chen, D. L. Donoho, M. A. Saunders, Atomic decomposition by Basis Pursuit, SIAM J. Sci. Comput. 20 (1999) no. 1, 33–61.

[9] A. R. Conn, N. I. M. Gould, P. L. Toint, Convergence of quasi-Newton matrices generated by the symmetric rank one update, Mathematical programming 50 (1991) 177–195.

[10] A. R. Conn, N. I .M. Gould, Ph. L. Toint, CUTE: constrained and unconstrained testing environment, ACM T. Math. Software. 21 (1995) 123–160.

[11] Y. H. Dai, J. Y. Han, G. H. Liu, D. F. Sun, H. X. Yin, Y. Yuan, Convergence properties of nonlinear conjugate gradient methods, SIAM J. Optim. 10 (1999) 345–358.

[12] Y. H. Dai, L. Z. Liao, R-linear convergence of the Barzilai and Borwein gradient method, IMA J. Numer. Anal. 22 (2002) 1–10.

[13] Y. H. Dai, Y. Yuan, A nonlinear conjugate gradient method with a strong global convergence property, IEEE SIAM J. Optim. 10 (1999) 177–182.

[14] E. D. Dolan, J. J. Moré, Benchmarking optimization software with performance profiles, Math. Program. 91 (2002) 201–213.

[15] D. L. Donoho, Compressed sensing, IEEE Trans. Inform. Theory. 52 (2006) 1289–1306.

[16] R. Fletcher, C. Reeves, Function minimization by conjugate gradients, Comput. J. 7 (1964) 149–154.

[17] S. Foucart, H. Rauhut, A Mathematical Introduction to Compressive Sensing, Applied and Numerical Harmonic Analysis, 2010.

[18] P. Gill, W. Murray, Quasi-Newton Methods for Unconstrained Optimization, IMA J. Appl. Math. 9 (1972) no. 1, 91–108.

[19] P. Gill, M. W. Leonard, Reduced-Hessian Quasi-Newton methods for unconstrained optimization, SIAM J. Optim. 12 (2001) no. 1, 209–237.

[20] A. Griewank, The 'global' convergence of Broyden-like methods with a suitable line search, J. of. Austral. Meth. Soc. Ser. B. 28 (1986) 75–92.

[21] L. Grippo, F. Lamparillo, S. Lucidi, A nonmonotone line search technique for Newtons method, SIAM J. Numer. Anal. 23 (1986) 707–716.

[22] W. W. Hager, H. Zhang, A new conjugate gradient method with guaranteed descent and an effcient line search, SIAM J. Optim. 16 (2005) 170–192.

[23] W. W. Hager, H. Zhang, A survey of nonlinear conjugate gradient methods, Pacific journal of Optimization 2 (2006) no. 1, 35–58.

[24] M. R. Hestenes, E. L. Stiefel, Methods of conjugate gradients for solving linear systems, J. Research Nat. Bur. Standards. 49 (1952) 409–436.

[25] J. Jian, Q. Chen, X. Jiang, Y. Zeng, J. Yin, A new spectral conjugate gradient method for large-scale unconstrained optimization, Optimization Methods and Software 32 (2017) no. 3, 503–515.

[26] M. Kimiaei, S. Ghaderi, A New Restarting Adaptive Trust-Region Method for Unconstrained Optimization, J. Oper. Res. Soc. China 5 (2017) 487–507.

[27] S. Min, L. Jing, A New Spectral Conjugate Gradient Method and Its Global Convergence, Journal of Information and Computing Science 8 (2013) 75–80.

[28] J. Nocedal, S. J. Wright, Numerical Optimization, Springer, NewYork, 2006.

[29] J. M. Ortega, W. C. Rheinboldt, Iterative Solution of Nonlinear Equations in Several Variables, Academic Press, New York, 1970.

[30] B. T. Polyak, The conjugate gradient method in extreme problems, USSR Comp. Math. Math. Phys. 9 (1969) 94–112.

[31] E. Polyak, G. Ribière, Note surla convergence de directions conjugées, Francaise Informat Recherche Opertionelle, 3e Année. 16 (1969) 35–43.

[32] J. M. Raydan, The Barzilai and Borwein gradient method for the large scale unconstrained minimization problem, SIAM J. Optim. 7 (1997) 26–33.

[33] K. Su, Z. X. Rong, A spectral conjugate gradient method under modified nonmonotone line search technique, Mathematica Aeterna 5 (2015) no. 4, 537–549.

[34] Z. J. Shi, J. H. Guo, A new trust region method for unconstrained optimization, Journal of Computational and Applied Mathematics. 213 (2008) 509–520.

[35] Y. Xiao, Q. Hu, Subspace Barzilai-Borwein gradient method for large-scale bound constrained optimization, Appl. Math. Optim. 58 (2008) 275–290.

[36] Z. Wana, Z. Yang, Y. Wang, New spectral PRP conjugate gradient method for unconstrained optimization, Applied Mathematics Letters. 24 (2011) 16–22.

[37] F. Wang, K. Zhang, C. Wang, L. Wang, A variant of trust-region methods for unconstrained optimization, Applied Mathematics and Computation 203 (2008) 297–307.

[38] G. Yu, L. Guan, W. Chen, Spectral conjugate gradient methods with sufficient descent property for large-scale unconstrained optimization, Optimization Methods and Software 23 (2008) 275–293.

[39] H. Zhang, W. W. Hager, A nonmonotone line search technique and its application to nuconstrained optimization, SIAM J. Optim. 14 (2014) no. 4, 1043–1056.

[40] Q. Zhou, F. Zhou, F. Cao, A nonmonotone trust region method based on simple conic models for unconstrained optimization, Applied Mathematics and Computation 225 (2013) 295–305.