URAL MATHEMATICAL JOURNAL, Vol. 11, No. 2, 2025, pp. 183–199

DOI: 10.15826/umj.2025.2.013

TWO MODIFIED CONJUGATE GRADIENT METHODS AND THEIR APPLICATION TO IMAGE RESTORATION PROBLEMS

Mehamdia Abd Elhamid

Université of M'hamed Bougara, Boumerdés, 35000, Algerie
a.mehamdia@univ-boumerdes.dz

Chaib Yacine

Mohamed Cherif Messaadia University, Souk Ahras, 41000, Algeria
y.chaib@univ-soukahras.dz

Hisham M. Khudhur

College of Computers Sciences and Mathematics, University of Mosul,
41003 Mosul, Nineveh Governorate, Iraq
hisham892020@uomosul.edu.iq

Raouf Ziadi

Université Ferhat Abbas Sétif 1,
Campus El Bez, Sétif 19000, Algeria
raouf.ziadi@univ-setif.dz

Abstract: Conjugate gradient methods have significantly contributed to the discovery of minimizes of large-scale unconstrained optimization problems. In this paper, based on the Liu–Storey conjugate gradient method, two modified conjugate gradient methods (named MC1 and MC2 methods) are presented for unconstrained optimization. Under usual assumptions, the two presented methods are proven to be sufficient descent at each iteration. The global convergence results of our methods is established using the strong Wolfe line search (SWLS). Numerical tests demonstrate the effectiveness of the MC1 and MC2 methods when compared to certain existing methods in view of the Dolan and Moré performance profile. Furthermore, the practical applications of these methods in image restoration problems is also considered.

Keywords: Unconstrained optimization, Conjugate gradient method, Analyse convergence, Numerical comparisons.

1. Introduction

Optimization plays a pivotal role in the domain of operational research, situated at the crossroads of computer science, mathematics, and economics. Its significance extends to applied mathematics, forming a basis for industry and engineering.

A nonlinear optimization problem contains an objective function that measures the performances or requirements of the system. Often, this function represents a profit, a time interval, a level, a sort of energy or a combination of different quantities that have a physical significance for the modeler. The objective function depends on some characteristics of the system, called variables or unknowns. The purpose of any optimization problem is to find the values of these variables that minimize (or maximize) the objective function subject to some constraints that the variables must satisfy.

Conjugate gradient (CG) methods are a popular family of iterative algorithms to solve large-scale nonlinear optimization problems due to appropriate features such as no need to calculate the second-order derivatives, low storage and computation, and suitable convergence rate. For more references on advances in conjugate gradient method, see A. E. Mehamdia et al. [17–21] and N. Andrei [2]. The CG algorithm is used in the fields of engineering optimization problems, neural net training, and image restoration. Hestenes and Stiefel formulated their algorithm for sets of linear systems of equations involving positive definite matrices. Initially, it was perceived as an optimization technique primarily designed for minimizing quadratic functions.

The role of the conjugate gradient method was expanded to include the optimization of non-quadratic functions in the 1964 by R. Fletcher and C. Reeves [11]. For nonquadratic functions, E. Polak and G. Ribiere [22] and B. T. Polyak [23] discussed the algorithms and the convergence research of different variations of conjugate gradient algorithms. Over the years, many variants of CG methods have been proposed, while some of them are widely used in practice, such as FR, PRP, DY, HS, CD, and LS. The PRP and LS methods have excellent performance in practical computation, because they posses an approximate restart feature when jamming occurs. however, their convergence properties are not is good.

If the line search is exact, then the LS method is identical to the PRP method. However, M. J. D. Powell [24] proposed a counter-example and showed the non-convergent sequence of the PRP method for the nonconvex function, which also applied to the LS method. To the best of our knowledge, the convergence of the standard LS method with various inexact searches is uncertain. Various modifications of the LS method have received growing interest.

For instance, Wei et al. [27] proposed hybrid method for solving unconstrained optimization problems. The method is based on the PRP method, denote it by WY. One important property of the WLY method is that it inherits the good properties of the PRP method, such as excellent numerical effect. Under the the strong Wolfe line search, H. Huang et al. [14] have proved that the WYL method satisfies the sufficient descent condition. and converges globally. In 2007, S. Yao et al. [25] extended this idea to the LS method and proposed a nonlinear CG method, we call the MLS method. Moreover, S. Yao et al. [25] proved that the MLS method also can produce sufficient descent direction and global convergence when SWLS is employed. Numerical results were given to support the theoretical results. Again, some more efficient LS type conjugate gradient method, G. Yu et al. [26], proposed a modified approach that generates a sufficient descent direction. The global convergence was proved for general function. Numerical experiments show that the LS1 method is more efficient when compared to other CG algorithms.

Continuing previous results, we propose two efficient conjugate gradient CG methods for solving unconstrained optimization problems. Under the SWLS, we establish the convergence properties of the MC1 and MC2 CG methods. Numerical results show that the two modifications are efficient, robust and each of these modifications outperforms four CG methods famous. Finally, an application of our methods in image restoration problems is also considered.

The rest of the paper is organized as follows. Some preliminaries are given in next section. In Section 3, we introduce the two modified methods and algorithms. In Section 4, we will present the sufficient descent condition and the global convergence proof of the two proposed methods. The numerical results and discussions are contained in Section 5. In Section 6, we focus an application of the new methods in image restoration problems. Conclusions and discussions are made in the last section.

2. Preliminaries

Consider the unconstrained optimization problem

$$\min\{f(x) : x \in \mathbb{R}^n\}, \tag{2.1}$$

where $f : \mathbb{R}^n \to \mathbb{R}$ is continuously differentiable and its gradient is denoted by

$$g_k = g(x_k) = \nabla f(x_k)$$

and $x_k \in \mathbb{R}^n$. Conjugate gradient methods are a class of important methods for solving the above problem often use the following iterative rules

$$x_{k+1} = x_k + \alpha_k d_k, \tag{2.2}$$

where x_k, is the current iteration point. The positive scalair α_k is called the steplength, usually determineted by a suitable inexact line search technique; such as the Wolfe line search conditions

$$\begin{cases} f(x_k + \alpha_k d_k) - f(x_k) \le \delta \alpha_k g_k^T d_k, \\ g_{k+1}^T d_k \ge \sigma g_k^T d_k. \end{cases} \tag{2.3}$$

or stronger version of the Wolfe line search given by

$$\begin{cases} f(x_k + \alpha_k d_k) - f(x_k) \le \delta \alpha_k g_k^T d_k, \\ \left|g_{k+1}^T d_k\right| \le \sigma \left|g_k^T d_k\right|. \end{cases} \tag{2.4}$$

where the parameters δ and σ satisfy $0 < \delta < \sigma < 1$, and d_k is a search direction computed by

$$d_{k+1} = -g_{k+1} + \beta_k d_k, \quad d_0 = -g_0, \tag{2.5}$$

and for descent methods, d_k is usually required to satisfy the sufficient descent property, if there exists a constant $c > 0$ such that

$$g_k^T d_k \le -c \|g_k\|^2, \quad \text{for all } k \geqslant 0.$$

Distinctive CG methods correspond to different choices for the conjugate gradient coefficient β_k, which in turn lead to quite diverse computational efficiency and convergence results of the corresponding methods. Well-known formulate for β_k are called the Fletcher–Reeves [11], Hestenes–Stiefel [13], Polak, Ribiere and Polyak [22, 23], Dai–Yuan [7], conjugate-descent [12] and Liu–Storey [15]. These parameters are given by the following formulae

$$\beta_k^{FR} = \frac{\|g_{k+1}\|^2}{\|g_k\|^2}, \quad \beta_k^{DY} = \frac{\|g_{k+1}\|^2}{y_k^T d_k}, \quad \beta_k^{CD} = \frac{\|g_{k+1}\|^2}{-g_k^T d_k},$$
$$\beta_k^{PRP} = \frac{g_{k+1}^T y_k}{\|g_k\|^2}, \quad \beta_k^{HS} = \frac{g_{k+1}^T y_k}{y_k^T d_k}, \quad \beta_k^{LS} = \frac{g_{k+1}^T y_k}{-g_k^T d_k},$$

where $\|\cdot\|$ denotes the Euclidean norm and

$$y_k = g_{k+1} - g_k.$$

For the general objective function LS, HS and PRP algorithms have good numerical performance, but DY, FR and CD methods have strong convergence property [6].

In order to prove the global convergence of the new methods, the following assumptions are required.

Assumption 1. *The level set*

$$S = \{x \in \mathbb{R}^n : f(x) \leq f(x_0)\},$$

is bounded, namely, there exists a constant $M > 0$, such that $\|x\| \leq M$, for all $x \in S$.

Assumption 2. *The objective function $f(x)$ is continuously differentiable in a neighborhood $\mathcal{N}$ of S and its gradient is Lipchitz continuous, namely, there is a constant $L > 0$, such that*

$$\|\nabla f(x) - \nabla f(y)\| \leq L\|x - yt\| \quad \textit{for all} \quad x, y \in \mathcal{N}.$$

By the sufficient descent property, the weak Wolfe line search (2.3) and Assumption 1, we declare that the sequence $\{x_k\}_{k\geq 0}$ is bounded. Combining this with Assumption 2, we know that there exists a positive constant $\Gamma \geq 0$, such that

$$\| \nabla f(x) \| \leq \Gamma, \quad \text{for all} \quad x \in \mathcal{N}. \tag{2.6}$$

It follows that Dai et al. [6] proved the sufficient condition for the convergence of CG methods with a strong Wolfe line search.

Lemma 1. *Let Assumptions 1 and 2 hold. Consider the method (2.2) and (2.5), where d_k is a descent direction, and α_k is obtained by the strong Wolfe line search. If*

$$\sum_{k\geq 0} \frac{1}{\|d_k\|^2} = \infty,$$

then

$$\lim_{k\to\infty} \inf \|g_k\| = 0.$$

3. Modified formulae and Algorithms

In this section, motivated by the structure of the LS method, we propose two hybrid CG methods to solve unconstrained optimization problems (2.1) and we focus on the MC1 and MC2 algorithms.

3.1. Main contributions

Recently, Wu [28] proposed an efficient CG method, where the search direction is yield

$$d_{k+1} = -g_{k+1} + \beta_k^{VLS} d_k, \quad d_0 = -g_0,$$

and

$$\beta_k^{VLS} = \frac{\|g_{k+1}\|^2 - t_k|g_{k+1}^T g_k|}{-\lambda d_k^T g_k + (1-\lambda)\max(0, g_{k+1}^T d_k t)},$$

where

$$t_k = \frac{\|g_{k+1}\|}{\|g_k\|}, \quad \lambda \in (0, 1).$$

Wu proved that the VLS method possesses sufficient descent conditions and global convergence properties when SWLS is employed [28].

Du et al. [9] improved the LS CG method and proposed CG method with good convergence and high efficiency, in which the conjugate parameters $\beta_k^{NVLS^*}$ are computed by

$$\beta_k^{NVLS^*} = \frac{\|g_{k+1}\|^2 - (|g_{k+1}^T g_k| \,/\, \|g_k\|^2) \cdot g_{k+1}^T g_k}{-g_k^T d_k}.$$

At every iteration, the NVLS* methods produce a sufficient descent condition and globally convergent if the SWLS is utilized. Recently, Zhu et al. [29] presented another new method which is called DDY1, where β_k in this method is given by

$$\beta_k^{DDY1} = \frac{\|g_{k+1}\|^2 - \mu_1 (g_{k+1}^T d_k)^2 |g_{k+1}^T g_k| \,/\, \left(\|g_k\| \cdot \|g_{k+1}\| \cdot \|d_k\|^2\right)}{y_k^T d_k},$$

where $\mu_1 \in [0,1]$. The authors proved that this method possesses sufficient descent conditions and global convergence properties when SWLS is employed [29].

The modified PRP type method will be described in this section. Moreover, the sufficient descent property of the algorithm is also given.

In this paper, we develop two modified LS-type CG methods (for short MC1 and MC2 methods). In the first method, we replace the formula

$$\frac{|g_{k+1}^T g_k|}{\|g_k\|^2} g_{k+1}^T g_k$$

in the numerator of $\beta_k^{NVLS^*}$ by $\rho_1 |g_{k+1}^T g_k| \omega_k$ with

$$\omega_k = \frac{|g_{k+1}^T d_k| g_{k+1}^T d_k}{\|g_{k+1}\| \cdot \|g_k\| \cdot \|d_k\|^2}.$$

Then the CG parameter (for short β_k^{MC1}) is taken as the form

$$\beta_k^{MC1} = \frac{\|g_{k+1}\|^2 - \rho_1 \left|g_{k+1}^T g_k\right| \omega_k}{-d_k^T g_k}, \tag{3.1}$$

where $\rho_1 \in [0,1]$. Applying hybridization to the conjugate parameter is an effective and reliable technique to improve the numerical performance and convergence properties. Therefore, motivated by [9, 28], we present our second method in the form of a hybrid method, where the conjugate parameter is defined by

$$\beta_k^{MC2} = \frac{\|g_{k+1}\|^2 - \rho_2 \max\left\{0, |g_{k+1}^T g_k| \cdot g_{k+1}^T g_k \,/\, \|g_k\|^2\right\}}{-d_k^T g_k + \max\left(0, g_{k+1}^T d_k\right)}, \quad \text{where} \quad \rho_2 \in [0,1]. \tag{3.2}$$

The primary attributes of these methods are as follows.

- A two modified CG algorithms, based on LS method, for solving unconstrained optimization are developed.
- The search direction generated by the presented methods is descent at each iteration with a strong Wolfe line search. Furthermore, this modifications are proved to be globally convergent under usual assumptions.
- Based on a great many numerical experiments, results demonstrate that our methods inherit the computation capacity of LS CG method and perform better compared to some existing methods for solving unconstrained optimization problems. Especially in CPU time, our methods are superior for the given test problems.
- The proposed methods are successfully applied to solving the image restoration problems less computation cost indicates the encouraging applicability of our approaches.

3.2. Algorithms

Now, based on the proposed conjugate parameters (3.1) and (3.2), the outlines of the corresponding algorithms are stated as follows.

Algorithm MC1.

Step 1. *Initializing.* Select positive constants

$$0 < \delta < \sigma < \frac{1}{1+\rho_1},$$

choose any initial point $x_0 \in \mathbb{R}^n$, let $d_0 = -g_0$.

Step 2. *Testing the iterations continuation.* If the $\|g_k\|_\infty \leq 10^{-6}$ is satisfied, then stops. Otherwise go to next step.

Step 3. *Line search.* Find the step length $\alpha_k > 0$ satisfying the strong Wolfe line search (2.4) and compute $x_{k+1} = x_k + \alpha_k d_k$.

Step 4. Calculate β_k by the formula (3.1).

Step 5. Compute the search direction d_k by using (2.5).

Step 6. Let $k = k+1$ and go to Step 2.

Algorithm MC2.

In Algorithm MC1

$$0 < \delta < \sigma < \frac{1}{1+\rho_1}$$

in Step 1 is replaced by

$$0 < \delta < \sigma < 1,$$

(3.1) in Step 4 is replaced by (3.2).

4. Convergence analyses

In this section, it is assumed that $g_k \neq 0$ for all $k \geq 0$, otherwise a stationary point is found. In order to obtain the convergence properties of the proposed methods.

This lemma to prove the convergence of our methods is also needed.

Lemma 2. *Let Assumptions 1 and 2 hold. If d_k is a descent direction and α_k satisfies the SWLS (2.4). Then*

$$\alpha_k \geq \frac{(1-\sigma)|g_k^T d_k|}{L\|d_k\|^2}. \tag{4.1}$$

P r o o f. See the proof of Lemma 2 in Lui and Li [16]. □

Remark 1. Assuming the beginning of this subsection and d_k is a descent direction, it is easy to obtain that $g_k^T d_k \neq 0$ for all $k \geq 0$. Thus, $\alpha_k = 0$ does not satisfy (2.4). This indicates that α_k obtained in the MC1 and MC2 methods is not equal to zero, i.e., there exists a constant $\lambda > 0$, such that

$$\alpha_k \geq \lambda, \quad \text{for all} \quad k \geq 0. \tag{4.2}$$

The following Lemma is needed to prove the sufficient descent property of the MC1 and MC2 methods.

Lemma 3. *The following inequality always holds*

$$\|g_{k+1}\|^2 - \rho_1|g_{k+1}^T g_k|\omega_k \leq (1+\rho_1)\|g_{k+1}\|^2, \quad \textit{with} \quad \rho_1 \in [0,1]. \tag{4.3}$$

P r o o f. Suppose that the ξ_k is the angle between the g_k and g_{k+1} vectors and the θ_k is the angle between the g_{k+1} and d_k vectors, then

$$\cos\xi_k = \frac{g_{k+1}^T g_k}{\|g_{k+1}\|\|g_k\|} \qquad \cos\theta_k = \frac{g_{k+1}^T d_k}{\|g_{k+1}\|\|d_k\|}.$$

So,

$$\begin{aligned}\|g_{k+1}\|^2 - \rho_1|g_{k+1}^T g_k|\omega_k &= \|g_{k+1}\|^2(1 - \rho_1|\cos\theta_k||\cos\xi_k|\cos\theta_k)\\ &\leq \|g_{k+1}\|^2(1+\rho_1\cos^2\theta_k|\cos\xi_k|)\\ &\leq (1+\rho_1)t\|g_{k+1}\|^2.\end{aligned}$$

The result can be achieved. □

4.1. Global convergence of MC1

The following theorem shows that the search direction generated by Algorithm MC1 satisfies the sufficient descent conditions.

Theorem 1. *Let the search direction d_k is yielded by the MC1 CG method. If the parameter σ satisfies*

$$0 < \sigma < \frac{1}{1+\rho_1},$$

then

$$g_k^T d_k \leq -c_1\|g_k\|^2, \quad \textit{for all} \quad k \geqslant 0. \tag{4.4}$$

P r o o f. The following proof is by induction. For

$$k = 0, \quad g_0^T d_0 = -\|g_0\|^2,$$

hence the sufficient descent condition holds for $k = 0$. Now, it is assumed that (4.4) holds for k and prove that for $k+1$.

From (3.1) and (4.4), we have

$$\beta_k^{MC1} = \frac{\|g_{k+1}\|^2\big(1-\rho_1|\cos\theta_k||\cos\xi_k|\cos\theta_k\big)}{-g_k^T d_k} \geq \frac{\|g_{k+1}\|^2(1-\rho_1)}{-g_k^T d_k} \geq 0. \tag{4.5}$$

Using (2.5) and (4.5), we get

$$g_{k+1}^T d_{k+1} = -\|g_{k+1}\|^2 + \beta_k^{MC1} g_{k+1}^T d_k \leq -\|g_{k+1}\|^2 + \beta_k^{MC1}|g_{k+1}^T d_k|. \tag{4.6}$$

By (2.4), (4.3) and (4.6), we obtain

$$g_{k+1}^T d_{k+1} \leq -\|g_{k+1}\|^2 + \frac{(1+\rho_1)\|g_{k+1}\|^2|g_{k+1}^T d_k|}{-g_k^T d_k} \leq -\|g_{k+1}\|^2 + \sigma(1+\rho_1)\|g_{k+1}\|^2.$$

It is concluded that

$$g_{k+1}^T d_{k+1} \leq -c_1\|g_{k+1}t\|^2,$$

where

$$c_1 = 1 - \sigma\left(1 + \rho_1\right),$$

so there is a constant $c_1 > 0$ with

$$\sigma < \frac{1}{1 + \rho_1}.$$

Therefore, the proof is completed. □

The following theorem establishes the global convergence of the MC1 method with the SWLS.

Theorem 2. *Consider that Assumptions 1 and 2 hold. Suppose the CG method yielded by (2.2) and (2.5), where the step size α_k satisfies the strong Wolfe line search condition (2.4) and parameter $\beta_k = \beta_k^{MC1}$, then this method is globally convergent, namely*

$$\lim_{k \to \infty} \inf \|g_k\| = 0. \tag{4.7}$$

P r o o f. Assume by contradiction that the formula (4.7) does not hold, there exists a constant $\gamma_1 > 0$, such that

$$\|g_k\| \geq \gamma_1, \quad \text{for all} \quad k \geq 0. \tag{4.8}$$

We have since the definition of β_k^{MC1} and (4.3)

$$\beta_k^{MC1} \leq \frac{(1 + \rho_1)\|g_{k+1}\|^2}{-d_k^T g_k}. \tag{4.9}$$

Hence, by (2.6), (4.4), (4.8) and (4.9), we have that

$$\beta_k^{MC1} \leq \frac{(1 + \rho_1)\Gamma^2}{c_1 \gamma_1^2} = E. \tag{4.10}$$

Thus, it follows from (2.5), (2.6), (4.2) and (4.10) that

$$\|d_{k+1}\| \leq \| g_{k+1} \| + \beta_k^{MC1} \frac{\| x_{k+1} - x_k \|}{\alpha_k} \leq M_1,$$

where

$$M_1 = \Gamma + E\frac{D}{\lambda},$$

here

$$D = \max\left\{\|y - z\| : \ y, z \in \mathcal{N}\right\}.$$

By taking the summation $k \geq 0$,

$$\sum_{k \geq 0} \frac{1}{\|d_k\|^2} = \infty.$$

So, applying Lemma 1, it is concluded that (4.7) is true. This is a contradiction with (4.8), so it is proved (4.7). □

4.2. Global convergence of MC2

Now, the sufficient descent direction of the MC2 method is proved.

Theorem 3. *Suppose the search direction sequence $\{d_k\}_{k\geq 0}$ is generated by Algorithm MC2. Then d_k satisfies the sufficient descent property*

$$g_k^T d_k \leq -c_2 \left\|g_k\right\|^2, \quad \textit{for all} \quad k \geqslant 0, \tag{4.11}$$

where $c_2 = 1 - \sigma$.

P r o o f. The following proof is by induction. For $k = 0$, $g_0^T d_0 = -\|g_0\|^2$, it is conclude that the sufficient descent condition holds for $k = 0$. Now, we assume (4.11) holds for k and prove that for $k + 1$.

With the definition of β_k^{MC2} and (4.11), then

$$\beta_k^{MC2} = \frac{\|g_{k+1}\|^2}{-d_k^T g_k + \max(0, g_{k+1}^T d_k)} \geq 0, \quad \text{if} \quad g_{k+1}^T g_k \leq 0,$$

and

$$\beta_k^{MC2} = \frac{\|g_{k+1}\|^2(1-\rho_2\cos^2\xi_k)}{-d_k^T g_k + \max(0, g_{k+1}^T d_k)} \geq \frac{\|g_{k+1}\|^2(1-\rho_2)}{-d_k^T g_k + \max(0, g_{k+1}^T d_k)} \geq 0 \quad \text{if} \quad g_{k+1}^T g_k \geq 0.$$

It is concluded

$$\beta_k^{MC2} \geq 0. \tag{4.12}$$

When having from (2.5), (3.2) and (4.12), so

$$g_{k+1}^T d_{k+1} = -\|g_{k+1}\|^2 + \beta_k^{MC2} g_{k+1}^T d_k \leq -\|g_{k+1}\|^2 + \beta_k^{MC2}|g_{k+1}^T d_k|. \tag{4.13}$$

By using (2.4), (3.2) and (4.12), then

$$g_{k+1}^T d_{k+1} \leq -\left\|g_{k+1}\right\|^2 + \frac{\left\|g_{k+1}\right\|^2}{-d_k^T g_k}\left|g_{k+1}^T d_k\right| \leq -(1-\sigma)\left\|g_{k+1}\right\|^2.$$

Above all, the conclusion of Theorem 3 holds. □

The following theorem to prove the global convergence of the MC2 method is used.

Theorem 4. *Assume that Assumption 1 and 2 hold. Let the sequence $\{x_k\}_{k\geq 0}$ be generated by the Algorithm MC2. If the stepsize α_k satisfies the Wolfe line search conditions, then*

$$\lim_{k\to\infty} \inf \|g_k\| = 0. \tag{4.14}$$

P r o o f. Suppose that (4.14) does not hold. Then there exists a constant $\gamma_2 > 0$, such that

$$\|g_k\| \geq \gamma_2, \quad \text{for all} \quad k. \tag{4.15}$$

In fact, from (2.6), (3.2), (4.11) and (4.15), we have

$$\beta_k^{MC2} \leq \frac{\|g_{k+1}\|^2}{-d_k^T g_k} \leq \frac{\Gamma^2}{c_2\gamma_2^2} = F. \tag{4.16}$$

Thus, it follows from (2.5), (2.6), (4.2) and (4.16), that

$$\|d_{k+1}\| \leq \| g_{k+1} \| + \beta_k^{MC2} \frac{\| x_{k+1} - x_k \|}{\alpha_k} \leq M_2,$$

where

$$M_2 = \Gamma + F\frac{D}{\lambda}.$$

Which implies that

$$\sum_{k \geq 0} \frac{1}{\|d_k\|^2} = \infty.$$

So, applying Lemma 1, it is concluded that (4.14) is true. This is a contradiction with (4.15), so it has been proved (4.14). □

5. Numerical experiments and discussions

In this section, some numerical experiments obtained with the two new proposed CG methods are presented. The test problems have been taken to [1, 3]. All the algorithms have been coded in MATLAB 2013 and compiler settings on the PC machine (2.5 GHz, 3.8 GB RAM) with Windows XP operating system. The computational results of MC1 method against the VLS* [9], NVLS* [9], DDY1 [29] and LS [15] methods are compared. On the other hand, the computational results of the MC2 method against the DPRP [8], NVLS* [9], LS1 [27] and LS [15] methods are compared. In this numerical result, all algorithms implement the SWLS condition with $\delta = 10^{-3}$ and $\sigma = 10^{-1}$. In this paper, the other parameters are specified by $\rho_1 = 0.8$ and $\rho_2 = 0.5$. The iteration is terminated if one of the following conditions is satisfied

(i) $\|g_k\|_\infty < 10^{-6}$, where $\|.\|_\infty$ is the maximum absolute component of a vector,
(ii) the number of iterations exceeded 2000,
(iii) the computing time is more than 500 s.

The performance difference is shown clearly between these methods MC1, MC2 and four conjugate gradient algorithms for each of them. The performance profile introduced by Dolan and Moré [10] is chosen to compare the performance according to the number of iterations and CPU time to rule as follows. Let S is the set of methods and P is the set of the test problems with n_p, n_s is the number of the test problems and the number of the methods, respectively. For each problem $p \in P$ and solver $s \in S$, denote $\tau_{p,s}$ be the number of iterations or CPU time required to solve problems $p \in P$ by solver $s \in S$. Then a comparison between different solvers based on the performance ratio is given by

$$r_{p,s} = \frac{\tau_{p,s}}{\min\{\tau_{p,i},\ 1 \leq i \leq n_s\}}.$$

Suppose that a parameter $r_M \geq r_{p,s}$ for all problems and solvers chosen, and $r_M = r_{p,s}$ if and only if solver s does not solve problem p.

The overall evaluation of the performance of the solvers is then given by the performance profile function given by

$$F_s(t) = \frac{\text{size}\{p:\ 1 \leq p \leq n_p,\ r_{p,s} \leq t\}}{n_p},$$

where $t \geq 1$ and

$$\text{size}\{p:\ 1 \leq p \leq n_p,\ r_{p,s} \leq t\}$$

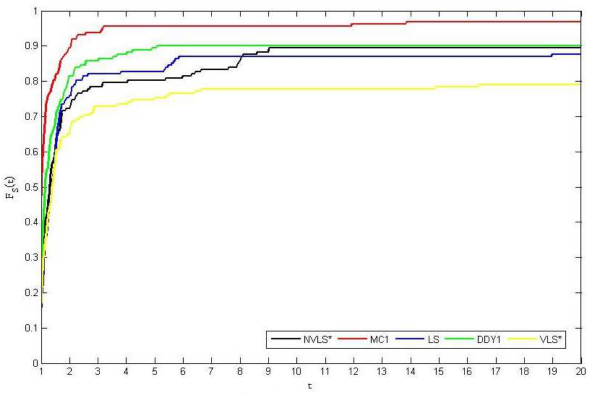

Figure 1. Performance profile on the CPU time.

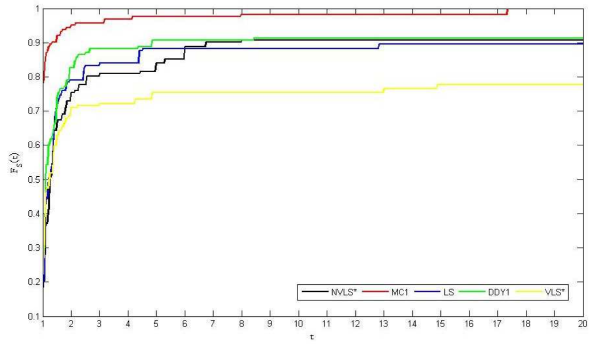

Figure 2. Performance profile on the number of iterations.

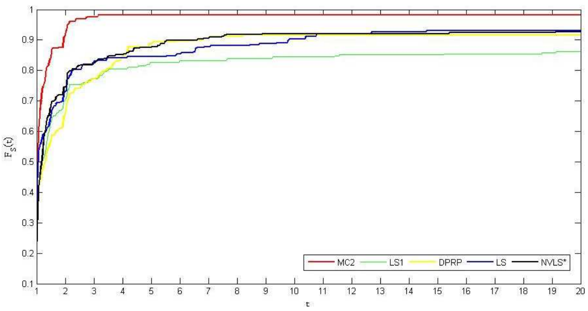

Figure 3. Performance profile on the CPU time.

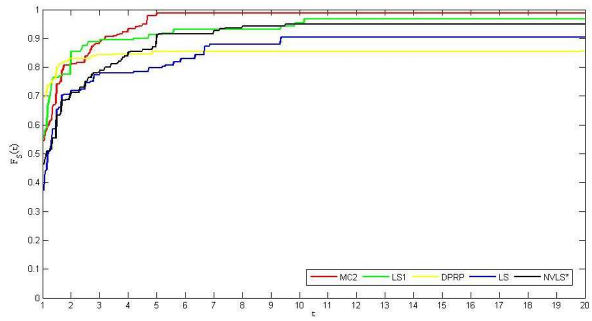

Figure 4. Performance profile on the number of iteration.

is the number of elements in the set $\{p: \ 1 \le p \le n_p, \ r_{p,s} \le t\}$. This function F_s: $[1, \infty[\to [0,1]$ is the distribution function for the performance ratio. The value of $F_s(1)$ is the probability that the solver will win the rest of the solvers.

In this numerical study, Dim denotes the dimension of the problem, ITR denotes the number of iterations, TIME denotes the CPU time and Inf denotes the algorithm failed to yield a solution for the problem.

Fig. 1 gives a performance comparison of the MC1 method versus VLS*, NVLS*, DDY1 and LS methods. As this figure indicates, the new Algorithm MC1 prevailed over all other methods, with respect to CPU time. This clearly confirms the effectiveness of the MC1 method. In particular, the NVLS* method outperforms the other methods except for the DDY1 method. It can be seen from Fig. 2 that the MC1 curve is mostly at the top of the VLS*, NVLS*, DDY1 and LS CG curves, so it is indicating that the MC1 Algorithm outperforms the other methods based on the number of iterations. Generally, the DDY1 method and the NVHS* method are better than the LS and VLS* methods.

From table in Figs. 5–6, it is clear that the average performance of the MC1, VLS*, NVLS*, DDY1 and LS CG methods are very similar to the results obtained from Figs. 1–2.

On the other side, Fig. 3 is the performance profile of the MC2, DPRP, NVLS*, LS1 and LS CG methods. From this figure, it is concluded that the MC2 method performs better than the DPRP, NVLS*, LS1 and LS CG methods, from the viewpoint of the CPU time. The NVLS* method behaves like the DPRP method, for the given test problems.

In Fig. 4, the number of iterations is used to compare the performance of the conjugate gradient

Function		MC1		DDY1		VLS*		LS		NVLS*	
	Dim	TIME	ITR	TIME	ITR	TIME	ITR	TIME	ITR	TIME	ITR
Engval 1	3000	0.6940	7	0.6570	7	0.5620	6	0.6250	7	0.5470	7
	10000	1.4690	5	1.4840	6	1.5000	6	1.9530	7	2.3440	7
Extended White And Holst	5000	1.1870	5	1.7820	6	1.2030	5	2.0780	6	1.2180	5
	5400	1.2660	5	1.5470	6	1.3440	5	1.4220	6	1.2780	5
Diagonal 4	12500	1.0930	12	1.1870	11	1.3120	12	1.1410	11	1.2030	11
	20000	1.7440	12	2.0710	12	2.0470	12	1.9380	12	2.0850	12
	30000	2.8340	12	2.8900	12	3.5410	12	2.9680	12	2.9060	12
Diagonal 2	1400	6.1890	424	6.8350	439	Inf	Inf	17.9250	928	65.1270	1987
	2000	10.3340	512	11.7270	558	Inf	Inf	28.9560	1021	Inf	Inf
	3100	24.5510	626	25.2250	632	Inf	Inf	104.4990	1577	Inf	Inf
Diagonal 1	140	0.8750	157	1.4400	227	Inf	Inf	4.3640	630	1.4510	227
	160	0.6560	141	1.4710	227	Inf	Inf	7.2660	955	1.4940	227
Chung	1400	8.8440	99	9.3910	106	9.4580	107	9.4530	106	9.4690	107
	1500	4.0000	34	10.3440	105	10.3280	106	7.7190	75	10.3290	106
	1900	9.9220	63	15.5630	102	14.2810	106	15.8140	107	14.5160	106
Quarticm	1060	13.4370	1084	13.7970	1088	Inf	Inf	13.6250	1081	13.7350	1083
	1050	13.2580	1084	13.5000	1083	Inf	Inf	13.4460	1080	13.5460	1085
	1100	14.3120	1091	14.6780	1102	Inf	Inf	15.3500	1098	14.6950	1105
Linear Perturbed	400	1.7650	215	1.8750	229	1.7970	27	1.8600	279	1.9210	235
	1600	16.5150	518	15.3570	479	15.1590	486	16.9700	489	15.8390	487
	2000	23.8950	577	23.7570	532	24.4470	558	48.0870	1004	23.6420	532
Fletcher	1200	0.2340	6	0.2660	7	0.2500	7	Inf	Inf	0.2660	7
	3000	0.7380	6	0.8440	7	0.7810	7	Inf	Inf	0.8130	7
	4000	0.9380	6	1.0780	7	0.9690	7	1.0780	7	1.1250	7
Liarwhd	300	17.3350	1546	Inf	Inf	Inf	Inf	18.5140	1549	19.3550	1701
Diagonal	2000	1.4060	29	2.6880	50	2.2240	39	2.4610	41	1.6720	36
	3500	3.1720	33	4.5770	40	4.6400	42	4.1440	41	3.6090	35
	4500	3.9070	34	6.0290	38	5.2080	35	4.4850	35	4.1870	33
	10000	10.0780	38	10.5610	37	10.4220	38	12.8280	45	10.1880	37
Cube	15600	0.1560	2	0.1880	2	0.2190	2	0.2340	2	0.1720	2
	1600	0.3120	5	0.3280	5	0.3910	5	0.3600	5	0.3280	5
TR-Sum of quadratics	6000	4.8280	24	5.7250	24	14.2910	67	5.0800	24	4.9440	24
	3980	3.2700	24	3.4560	24	9.7570	66	3.3870	24	3.3090	24
	3200	2.8230	22	2.9390	24	8.7400	66	2.9390	24	2.8920	24
Griewank	3200	30.8750	836	32.1010	836	31.5540	836	31.2470	836	31.1010	836
	3600	33.2660	829	34.1880	829	33.4060	829	33.2820	829	34.1270	829
	5000	49.9620	879	50.8990	879	51.1890	879	52.2810	879	50.8900	879
Exponential	601	0.0310	2	0.0470	3	0.0310	4	0.0310	5	0.0310	4
Extended Hiebert	1600	1.5000	39	1.5940	44	Inf	Inf	2.2340	78	1.7840	48
	1750	2.4060	57	2.7650	65	Inf	Inf	4.1870	127	2.5780	63
	1800	1.7570	44	1.7150	40	Inf	Inf	1.7970	57	1.8430	44
	2600	2.3280	42	2.4290	44	Inf	Inf	2.2650	55	2.6250	48
Diagonal DiagAup1	1980	0.0790	4	0.0940	4	0.1560	4	0.1560	4	0.0930	4
Double Border Arrow Up	2200	2.9550	55	3.4630	62	3.3720	62	3.3720	62	3.5630	68
	3000	1.3750	29	1.5780	36	1.9220	41	1.9220	41	2.4920	50
	4500	3.4140	32	3.8440	33	3.9220	35	3.9220	35	4.8910	38
Almost Perturbed Quartic	1300	9.0940	276	4.7550	142	5.0010	221	5.2500	229	5.0160	223
	1600	2.7820	143	2.6090	140	2.7180	142	2.7720	143	2.7110	142
	2500	4.3510	150	4.1580	148	4.2520	150	4.2570	150	4.2580	151
Almost PerQuadratic	300	9.1210	649	16.3160	1129	Inf	Inf	Inf	Inf	29.2580	1171
Arwhead	1280	0.1660	5	0.0470	2	0.1720	5	Inf	Inf	0.0470	2

Figure 5. The simulation results of MC1, DDY1, VLS*, LS and NVLS* methods.

codes MC2, DPRP, NVLS*, LS1 and LS CGMs. Relative to this metric, MC2 achieves the top performance, followed by LS1, then NVLS*, then LS and DPRP.

From table in Fig. 7 and Fig. 8, it is clear that the average performance of the MC2, DPRP, NVLS*, LS1 and LS CG methods are very similar to the results obtained from Fig. 3 and Fig. 4.

6. Application to image restoration problems

The image restoration problem is considered as one of the most challenging optimization problems. In this section, we will use the MC1 and MC2 methods to restore images corrupted by

Function		MC1		DDY1		VLS*		LS		NVLS*	
	Dim	TIME	ITR	TIME	ITR	TIME	ITR	TIME	ITR	TIME	ITR
Sumsquares	1000	3.8220	71	3.1850	65	3.9654	73	2.9680	66	2.4540	1300
	1900	3.2760	71	3.3130	74	3.7640	73	3.5780	79	3.7810	1600
	2400	3.8280	78	3.9070	78	4.7370	85	3.9220	78	4.6270	88
Sphere	1600	0.0890	10	0.0940	9	Inf	Inf	0.0930	9	0.0940	9
	4000	0.4840	10	0.5470	11	Inf	Inf	0.5470	11	0.4880	10
	8000	0.9590	11	0.9680	12	Inf	Inf	1.0730	12	1.0320	12
Schwefel 223	2900	0.5150	4	0.5160	4	1.0470	7	Inf	Inf	0.2500	4
	9000	0.8430	6	1.1960	6	0.8430	6	0.8430	6	1.0510	6
Rosenbrock	3000	2.0650	51	1.9630	45	3.8250	83	3.9520	88	4.7800	100
	3500	1.7180	29	2.1720	40	3.4840	69	2.7520	58	3.1560	60
	4000	5.7810	69	3.2340	40	6.3280	96	8.0280	89	8.7960	102
	6000	2.6560	29	4.5770	50	7.1400	69	7.6240	72	10.6500	83
Raydan 2	2800	Inf	Inf	8.9470	89	Inf	Inf	17.5780	170	5.0940	54
	3400	1.5160	18	1.4840	16	Inf	Inf	1.5630	18	1.0000	12
	4600	4.7810	31	169.674	812	Inf	Inf	54.1200	299	12.5940	83
Dixon	2000	0.3850	12	0.4540	12	0.3860	12	Inf	Inf	0.4700	13
Rastrigin	900	0.4840	19	1.4370	51	0.9680	35	2.0000	70	0.9530	35
	1100	1.2570	38	1.7500	52	1.3910	43	1.3910	43	1.3750	43
	1200	0.8290	22	2.1710	63	1.4690	43	1.4690	43	1.4690	43
Qing	600	5.6480	243	5.6750	259	16.9050	608	15.0620	619	14.7470	608
	1000	12.1720	335	12.2500	338	37.7970	940	25.2410	655	37.2960	940
	2000	38.8590	494	35.1350	498	118.009	1395	117.014	1378	118.938	1395
penalty	1000	0.8120	34	1.8280	92	1.8280	63	2.2030	77	1.6560	60
	1500	0.8130	24	1.8910	50	1.1870	31	4.8750	117	0.9370	27
	1950	1.2030	29	27.2340	461	4.0780	83	7.2030	139	3.7810	77
Himmelblau	10000	4.7190	15	8.4380	21	5.0160	15	11.7190	30	5.2810	15
	18000	4.7000	15	8.0710	21	4.9220	15	11.1340	30	5.1020	15
	20000	6.1130	16	8.9840	19	6.9150	21	6.9150	16	6.5000	16
Hager	900	2.9260	98	4.8830	155	7.1040	222	Inf	Inf	7.0620	221
	1090	2.9510	97	4.6780	137	63.1820	1070	Inf	Inf	84.7320	1558
	2000	18.931	266	17.5690	263	134.6840	1327	Inf	Inf	118.9310	1327
Zakharov	1000	0.1090	3	0.1250	4	0.1090	3	0.1560	4	0.1250	3
	2800	0.2500	3	0.3750	4	0.2970	3	0.4220	4	0.2660	3
	6000	0.6130	3	0.7340	4	0.8060	3	0.7810	4	0.7340	3
	10000	1.0000	3	1.3910	4	0.9220	3	1.4220	4	1.0310	3
Hmmelbh	800	17.1800	431	18.2490	454	18.5070	456	57.5680	1227	76.4820	1566
Liarwhd	4500	0.0310	2	0.0470	2	0.0470	2	0.0470	2	0.0630	2
	1500	0.0150	2	0.0310	2	0.0310	2	0.0150	2	0.0320	2
Harkerp	1000	6.7890	206	6.8510	207	10.4670	276	6.8660	207	6.8960	207
	1200	5.5310	188	5.5470	190	5.5470	190	5.5520	190	5.5000	190
Prod 1	1800	0.8840	9	1.3900	11	48.1240	314	1.0780	11	1.1090	11
	2800	1.7220	8	2.1870	10	2.1800	10	2.1880	10	2.1720	10
	3000	1.7180	11	1.9540	11	1.8750	11	1.9060	11	1.8910	11

Figure 6. The simulation results of MC1, DDY1, VLS*, LS and NVLS* methods (Cont.).

impulse noise.

Chan et al. [5] employ a two-phase scheme to repair images affected by impulse noise. In the first phase, a median filter was employed to identify noise pixels. Let X represent the original image corrupted by salt and pepper noise, with dimensions $n \times m$, let $A = \{1, 2, ..., n\} \times \{1, 2, ..., m\}$ be the index set of the image X, the set of indices for the noise pixels is represented by $N \subset A$ where $|N|$ denotes the total number of elements within N, $P_{i,j}$ is the collection of four nearest neighbors of the pixel at $(i, j) \in A$. In the second phase, noisy pixels are cleaned by solving the nonsmooth minimization problem

$$f_\alpha(v) = \sum_{(i,j)\in N} \left[|v_{i,j} - z_{i,j}| + \frac{\xi}{2} t(2S^1_{i,j} + S^2_{i,j}) \right], \tag{6.1}$$

where a column vector of length $|N|$ is indicated by $v_{i,j} = [v_{i,j}]_{(i,j)\in\mathbb{N}}$, ξ denotes a regularization parameter and

$$S^1_{i,j} = 2 \sum_{(m,\, n)\in P_{i,j}\cap N^c} \phi_\alpha\left(v_{i,j} - z_{m,n}\right) \quad \text{and} \quad S^2_{i,j} = \sum_{(m,\, n)\in P_{i,j}\cap N} \phi_\alpha\left(v_{i,j} - z_{m,n}\right).$$

Here,

$$\phi_\alpha(t) = \sqrt{\alpha + t^2}$$

Function		MC2		NVLS*		LS		LS1		DPRP	
	Dim	TIME	ITR	TIME	ITR	TIME	ITR	TIME	ITR	TIME	ITR
Almost perturbed Quartic	2000	7.9540	95	8.2180	233	8.2180	232	8.7490	233	7.2660	224
	5000	3.7720	21	17.2360	246	17.0860	240	16.3540	237	14.5350	239
	6000	10.3970	27	51.3940	286	32.5190	250	36.8020	218	21.9130	251
Cube	2400	0.8020	5	0.8440	5	0.8440	5	0.8120	5	0.8220	5
	3000	1.0940	5	1.2340	5	1.6250	5	1.5780	5	1.5830	5
	4000	1.6000	5	1.5360	5	1.8230	5	1.6100	5	1.6340	5
ExtendedWhit eand Holst	4000	1.0000	5	1.5580	7	1.1300	5	1.2400	5	1.5600	6
	6000	1.2600	5	1.8590	7	1.2110	5	1.3970	5	1.5280	6
	7000	1.7090	5	2.5480	7	1.7010	5	1.9670	5	2.3950	6
Rastrigin	900	0.4250	18	1.7480	53	0.3900	16	1.4690	53	1.4690	53
	1200	1.8060	51	1.6870	49	1.7970	51	0.2970	8	1.6880	49
	1400	2.1220	54	1.7500	44	inf	inf	inf	inf	inf	inf
Staircase S3	300	0.5940	401	0.9370	616	0.6410	401	0.8020	217	0.6090	117
power	700	25.1120	1080	29.0750	1191	27.1890	1141	31.9320	1097	Inf	Inf
	400	6.8640	530	8.1690	611	8.1900	609	10.3410	555	8.1480	615
Double Border Arrow Up	2000	0.4250	3	0.2410	4	0.2030	4	0.1720	4	0.1400	3
Diagonal	4770	1.9060	18	4.3120	38	4.1060	34	5.2140	20	5.1020	36
	5000	2.4040	18	5.3250	41	4.9470	36	5.7010	22	4.6990	36
	6000	2.7300	18	7.8130	45	4.7610	32	6.7460	31	6.2400	40
Schwefel1223	1000	0.1220	5	0.1410	5	0.1250	5	0.1410	5	0.1560	5
	3000	0.2650	5	0.5950	5	0.2740	5	0.2350	5	0.3440	5
	4000	0.3320	5	0.6210	5	0.3750	5	0.3600	5	0.4730	5
Liarwhd	110	0.9530	174	Inf	Inf	Inf	Inf	Inf	Inf	1.0150	175
penalty	2550	2.0630	30	3.4340	48	14.7100	169	2.1040	34	5.4650	70
	850	0.0630	8	0.1010	9	0.3380	14	0.1090	8	0.4440	29
Qing	800	7.9190	287	23.3570	732	7.9540	287	38.2050	578	20.0570	628
	992	10.6720	321	17.3590	499	11.8590	354	60.4680	395	38.2970	977
Feletcher	790	0.0310	3	0.0630	4	0.0310	3	0.0320	3	0.0630	4
	1950	0.1400	2	0.1730	3	0.1410	2	0.1410	2	0.1710	3
Rosenbrock	2200	2.4310	84	2.6850	83	0.5000	17	2.9060	84	2.4590	74
	3500	2.8400	47	4.5720	60	4.5810	63	6.5470	90	6.3520	94
	4500	6.2350	86	6.4550	84	5.3590	73	4.9090	64	6.5510	86
Raydan2	1800	0.0470	2	0.1580	4	0.1560	5	0.1570	4	0.1570	4
	2200	0.1100	7	2.0920	36	1.3910	31	0.2120	7	0.2350	8
Styblinski	600	1.5160	28	2.2450	49	Inf	Inf	2.0940	48	1.2030	28
	800	1.5460	31	1.7370	31	Inf	Inf	1.6670	28	2.7120	60
	900	3.1560	63	4.3040	84	Inf	Inf	3.3460	64	3.3530	65
Tridiagonal	100	2.2030	505	3.7810	894	2.1720	505	3.4220	683	3.3770	683
	200	12.140	1133	18.453	1826	12.129	1133	Inf	Inf	Inf	Inf
	300	35.909	1940	Inf	Inf	34.859	1940	Inf	Inf	Inf	Inf
Extended Hiebert	1400	1.9250	54	2.2030	64	3.7660	141	5.2030	194	4.0630	157
	1850	2.6110	57	2.9220	65	3.5470	104	7.2500	68	8.1580	231
	2000	2.6870	53	3.5940	68	51.5270	769	8.5740	221	9.2970	222
Tridia	60	0.0940	30	0.3130	90	Inf	Inf	Inf	Inf	Inf	Inf
	80	0.1250	30	0.1560	38	Inf	Inf	Inf	Inf	Inf	Inf
	100	0.1570	30	0.1410	27	Inf	Inf	Inf	Inf	Inf	Inf
Perturbed	1000	10.6870	520	13.2280	593	12.4220	592	11.7450	536	13.6660	536
	1500	19.6370	741	20.1090	746	19.6970	741	25.9530	746	27.4300	746
Prod 2	1500	0.1880	5	Inf	Inf	0.1950	6	0.2310	6	0.2180	6
	1800	0.2500	5	Inf	Inf	0.2610	6	0.2840	6	0.2720	6
Almost Perturbed Quadra! c	800	8.8910	329	Inf	Inf	9.3450	333	9.8850	375	9.6420	375
	1600	27.0470	541	29.7190	584	27.1750	541	24.0310	565	25.0930	510

Figure 7. The simulation results of MC2, NVLS*, LS, LS1 and DPRP methods.

is an edge-protecting function with parameter $\alpha > 0$,

$$N^c = \{(i, j) \in A / z_{m,n} = z_{i,j}, \text{ and } z_{i,j} = S_{min} \text{ or } S_{max}\}.$$

In order to minimize calculation costs and save time when solving (6.1), the nonsmooth term was eliminated by Cai et al. [4], it introduces the following, unconstrained optimization

$$f_\alpha(v) = \sum_{(i,j)\in N} [2 \times S^1_{i,j} + S^2_{i,j}]. \tag{6.2}$$

In this study, we are particularly interested in how quickly the miniaturization problem (6.2) is resolved. The quality of the recovered images is measured using the peak signal-to-noise ratio (PSNR), which is defined as

$$PSNR = 10 \log_{10} \frac{255^2}{1/MN \cdot \sum_{i,j} (v^r_{i,\ j} - v^*_{i,j})^2},$$

Function		MC2		NVLS*		LS		LS1		DPRP	
	Dim	TIME	ITR	TIME	ITR	TIME	ITR	TIME	ITR	TIME	ITR
Nonscomp	1500	1.1090	18	1.2340	20	1.1410	19	3.1560	19	1.2500	20
	2100	1.6940	18	1.5780	19	1.5310	19	4.1880	19	1.5780	19
	5000	8.4690	35	8.5990	36	8.7810	36	12.9370	36	8.7030	36
Quarticm	1000	10.4200	853	12.9650	1033	12.8340	1032	13.1080	1033	12.4950	1022
	2400	16.1090	1132	19.9840	1130	19.4530	1130	20.0750	1130	19.7910	1191
Raydan1	80	0.1620	90	0.0780	55	0.0780	55	0.1090	54	0.1090	88
	125	0.1870	65	0.2030	77	0.2660	77	0.2970	77	0.3130	155
	2000	0.4370	17	0.4370	17	0.3130	14	0.2970	14	0.3230	14
Diagonal2	1000	16.0210	1019	Inf	Inf	16.4090	1070	Inf	Inf	Inf	Inf
	1200	9.0620	647	Inf	Inf	9.5760	669	Inf	Inf	Inf	Inf
	1400	14.1890	712	Inf	Inf	15.0970	720	Inf	Inf	Inf	Inf
Diagonal4	1900	0.1310	10	0.1330	10	0.1110	10	0.1090	10	0.1110	10
	3000	0.3400	11	0.3300	10	0.1870	10	0.2640	10	0.2030	10
	4000	0.2940	10	0.2810	10	0.2500	10	0.2810	10	0.2190	9
Exponetial	601	0.0320	3	0.0320	3	0.0780	5	0.0470	2	0.0320	3
Griewank	1700	3.2900	158	3.9120	164	3.9090	164	3.8920	162	3.9090	164
	2000	5.2750	140	5.3750	140	5.4690	140	5.3430	139	5.3440	139
	2800	9.7874	176	9.4620	175	9.8840	176	11.0390	176	9.9210	176
Hager	1000	3.2920	103	10.0490	279	3.4540	119	10.9450	119	8.4600	340
	1200	16.4280	351	9.7810	238	4.1330	119	18.3300	119	11.1930	450
	3000	13.4840	281	10.7370	213	13.0150	248	71.6770	240	18.4820	480
Himmelblau	1400	0.3440	18	0.3910	19	0.3590	19	0.4680	19	0.3750	19
	1500	0.3120	17	0.4060	19	0.3900	18	0.3940	19	0.4380	19
	2000	0.4530	18	0.5310	19	0.5160	19	0.5000	19	0.5310	19
Chung	1000	4.2960	93	4.6870	104	4.5000	104	4.7190	104	4.7190	104
	1100	5.3240	13	5.5900	106	5.6840	104	5.6320	106	5.6360	106
	1200	5.6800	85	6.5090	105	6.0730	99	6.5090	105	6.4340	105
Harkerp	2000	0.7030	13	0.5940	12	0.6100	12	0.2810	7	0.5780	12
Quadratic	1000	0.1720	6	18.7370	823	0.6720	23	0.2250	6	40.1560	1389
	2000	6.1720	188	5.9840	163	1.0620	18	6.7190	188	20.1090	524
	3500	1.2660	6	84.3590	906	2.6880	25	0.6090	6	99.4250	1045
	4000	0.6880	6	140.3140	958	2.9060	25	0.6250	6	156.6710	1113
Quartic	805	0.6100	60	0.9220	91	0.9690	89	0.9180	90	0.9680	89
	950	1.3990	128	2.2030	194	2.2030	195	1.9230	188	2.6220	195
	1400	2.4360	129	3.0980	199	3.1520	199	3.0980	199	3.8350	199
	1800	2.8910	132	4.4890	200	4.3910	200	4.4890	200	4.4810	203
Sphere	1000	0.1050	11	0.1410	14	0.1400	14	0.1560	15	0.1410	14
	2600	0.1440	9	0.1690	10	0.1650	10	0.3720	10	0.1570	10
	5000	0.5150	19	0.8590	21	0.8280	21	0.8430	21	0.8750	22
Sumquares	1800	4.4630	78	4.5730	78	4.3430	79	4.6030	78	5.1620	93
	2300	5.5320	86	5.7810	89	5.7660	88	5.7540	89	8.3760	127
	2900	9.2150	101	9.5620	100	9.9530	106	9.5700	100	9.9510	105
Arwhed	1800	0.1720	5	0.1410	5	0.1560	5	0.1560	5	0.1720	5
	2000	0.2180	5	0.2650	5	0.2660	5	0.3120	5	0.2650	5
	3000	0.4530	5	0.4840	5	0.5320	5	0.5630	5	0.5000	5

Figure 8. The simulation results of MC2, NVLS*, LS, LS1 and DPRP methods (Cont.).

where $v^r_{i,j}$ and $v^*_{i,j}$ indicate the pixel values of the original and restored images, respectively. All the compared methods terminate if one of the following criteria is satisfied,

$$\frac{|f(v_{k+1}) - f(v_k)|}{|f(v_{k+1})|} \leq 10^{-4} \quad \text{and} \quad \|f(v_{k+1})\| \leq 10^{-4}(1 + |f(v_{k+1})|).$$

Corresponding results are presented in table in Fig. 9. It includes the PSNR (peak signal-to-noise ratio), the CPU time (TCPU) and the number of iterations (ITR). The MC1 and MC2 methods outperform the NVLS* and LS methods in terms of the number of iterations, CPU time and peak signal-to-noise ratio, as shown in table in Fig. 9.

7. Conclusion

Based on some famous previous CG methods, this paper has developed two new conjugate gradient methods. Under the strong Wolfe line search condition, the global convergence properties and the sufficient descent condition of the MC1 and MC2 methods have been established. Relying on many numerical experiments, results demonstrate that these methods are very robust and effective and perform better compared to some existing methods in minimizing some unconstrained optimization problems. These can be seen more clearly in Figs. 1–4 and Tables 1–2. The MC1 and MC2 methods were extended to solve image restoration problems. From the numerical experiments in Tables 3, it is noted that the efficiency of the MC1 and MC2 methods in solving image restoration problems is measured by the number of iterations, and CPU time by comparison with various conjugate gradient methods.

		MC1			MC2			NVHS*			LS		
Image	**Noise ratio**	**ITR**	**TCPU**	**PSNR**	**ITR**	**TCPU**	**PSNR**	**ITR**	**TCPU**	**PSNR**	**ITR**	**TCPU**	**PSNR**
baboon	%30	9	2.16	18.99	45	2.99	23.28	23	3.96	23.28	18	3.02	23.28
	%50	20	3.86	14.67	77	6.60	21.85	4	2.74	11.87	4	2.80	11.82
	%70	13	4.04	13.25	106	10.72	20.42	31	9.86	20.41	19	8.29	20.39
	%90	14	5.62	12.50	83	10.99	18.73	26	11.77	18.63	19	9.92	17.59
barbara	%30	7	7.61	20.64	65	15.85	28.40	22	12.37	28.41	15	14.09	28.27
	%50	7	11.82	15.32	88	31.18	26.36	32	28.71	26.35	16	16.50	26.13
	%70	14	23.12	13.50	107	47.13	24.39	34	50.98	24.38	24	40.13	24.37
	%90	5	22.52	11.05	110	65.25	22.24	39	56.50	22.19	30	62.59	22.09
Brain	%30	17	2.47	20.59	69	3.89	29.73	17	2.76	29.71	21	3.74	29.75
	%50	11	2.92	15.69	91	7.46	28.25	30	7.74	28.04	14	4.94	23.54
	%70	20	5.12	13.90	110	10.95	26.29	21	7.96	25.67	6	5.08	9.92
	%90	16	6.81	10.68	114	14.09	22.50	46	12.71	22.26	45	21.18	22.24
cat8	%30	10	3.86	16.47	76	7.76	23.19	26	6.65	19.84	15	4.78	19.32
	%50	13	4.54	13.37	80	9.56	19.03	38	10.10	15.87	3	3.44	10.28
	%70	13	8.32	9.52	92	13.85	15.50	37	13.63	12.57	3	5.20	8.99
	%90	8	10.26	7.67	111	19.62	12.38	28	15.61	9.64	19	16.10	9.40
Lena	%30	9	9.57	22.15	62	16.29	37.00	20	13.45	36.61	18	13.38	37.02
	%50	15	15.40	15.47	85	30.85	34.48	35	31.42	34.46	18	26.65	32.20
	%70	14	32.23	13.66	105	46.75	31.24	35	51.70	31.06	19	31.34	30.92
	%90	14	35.92	13.13	115	75.10	26.35	30	52.72	26.07	11	40.09	18.23
zainb1	%30	8	10.81	22.84	65	18.88	32.31	25	19.30	32.29	17	14.05	32.24
	%50	13	20.36	15.88	91	42.35	30.19	32	33.51	30.11	20	21.42	29.88
	%70	10	15.64	14.11	117	56.00	27.64	33	56.45	27.43	19	31.24	26.85
	%90	6	20.19	9.78	118	64.05	24.27	3	16.38	9.07	25	53.98	22.61

Figure 9. Numerical results of FR, DY, NFR, NDY algorithms.

Acknowledgements

The authors are very grateful to the anonymous referees for their valuable comments and useful suggestions, which improved the quality of this paper.

REFERENCES

1. Andrei N. An unconstrained optimization test functions collection. *Adv. Model. Optim.*, 2008. Vol. 10, No. 1. P. 147–161.
2. Andrei N. *Nonlinear Conjugate Gradient Methods for Unconstrained Optimization.* Ser. Springer Optim. Appl. Cham: Springer, 2020. 498 p. DOI: 10.1007/978-3-030-42950-8
3. Bongartz I., Conn A. R., Gould N., Toint Ph. L. CUTE: Constrained and unconstrained testing environment. *ACM Trans. Math. Software*, 1995. Vol. 21, No. 1. P. 123–160.
4. Cai J.-F., Chan R., Morini B. Minimization of an edge-preserving regularization functional by conjugate gradient type methods. In: *Image Processing Based on Partial Differential Equations. Tai XC., Lie KA., Chan T.F., Osher S. (eds.).* Ser. Math. Vis. Berlin, Heidelberg: Springer, 2007. P. 109–122. DOI: 10.1007/978-3-540-33267-1_7
5. Chan R. H., Chung-Wa Ho, Nikolova M. Salt-and-pepper noise removal by median-type noise detectors and detail preserving regularization. *IEEE Trans. Image Process.*, 2005. Vol. 14, No. 10. P. 1479–1485. DOI: 10.1109/TIP.2005.852196
6. Dai Y. H., Han J. Y., Liu G. H., Sun D. F., Yin X., Yuan Y. Convergence properties of nonlinear conjugate gradient methods. *SIAM J. Optim.*, 1999. Vol. 10, No. 2. P. 348–358. DOI: 10.1137/S1052623494268443
7. Dai Y. H., Yuan Y. A nonlinear conjugate gradient method with a strong global convergence property. *SIAM J. Optim.*, 1999. Vol. 10, No. 1. P. 177–182. DOI: 10.1137/S1052623497318992

8. Dai Z., Wen F. Another improved Wei–Yao–Liu nonlinear conjugate gradient method with sufficient descent property. *Appl. Math. Comput.*, 2012. Vol. 218, No. 14. P. 7421–7430. DOI: 10.1016/j.amc.2011.12.091
9. Du X. W., Zhang P., Ma W. Some modified conjugate gradient methods for unconstrained optimization. *J. Comput. Appl. Math.*, 2016. Vol. 305. P. 92–114. DOI: 10.1016/j.cam.2016.04.004
10. Dolan E. D., Moré J. J. Benchmarking optimization software with performance profiles. *Math. Program.*, 2002. Vol. 91, No. 2. P. 201–213. DOI: 10.1007/s101070100263
11. Fletcher R., Reeves C. Function minimization by conjugate gradients. *Comput. J.*, 1964. Vol. 7, No. 2. P. 149–154. DOI: 10.1093/comjnl/7.2.149
12. Fletcher R. *Practical Methods of Optimization*, 2nd ed., New York: Wiley, 1987. 450 p.
13. Hestenes M. R., Stiefel E. L. Methods of conjugate gradients for solving linear systems. *J. Research Nat. Bur. Standards*, 1952. Vol. 49, No. 6. P. 409–436.
14. Huang H., Wei Z., Yao S. The proof of the sufficient descent condition of the Wei–Yao–Liu conjugate gradient method under the strong Wolfe–Powell line search. *Appl. Math. Comput.*, 2007. Vol. 189, No. 2. P. 1241–1245. DOI: 10.1016/j.amc.2006.12.006
15. Liu Y., Storey C. Efficient generalized conjugate gradient algorithms, part 1: Theory. *J. Optim. Theory Appl.*, 1991. Vol. 69. P. 129–137. DOI: 10.1007/BF00940464
16. Liu J. K., Li S. J. New hybrid conjugate gradient method for unconstrained optimization. *Appl. Math. Comput.*, 2014. Vol. 245. P. 36–43. DOI: 10.1016/j.amc.2014.07.096
17. Mehamdia A. E., Chaib Y., Bechouat T. Two modified conjugate gradient methods for unconstrained optimization and applications. *RAIRO Oper. Res.*, 2023. Vol. 57. P. 333–350. DOI: 10.1080/10556788.2024.2400710
18. Mehamdia A. E., Khudhur H. M. A globally convergent of two conjugate gradient methods with application to image restoration problems. *Numer. Algebra Control Optim.*, 2025. Vol. 15, No. 3. P. 645–660. DOI: 10.3934/naco.2024027
19. Mehamdia A. E., Chaib Y., Bechouat T. Two improved nonlinear conjugate gradient methods with application in conditional model regression function. *J. Ind. Manag. Optim.*, 2025. Vol. 21, No. 1. P. 658–675. DOI: 10.3934/jimo.2024098
20. Mehamdia A. E., Chaib Y., Bechouat T. Improved conjugate gradient methods and application to nonparametric estimation. Appl. Math. (Warsaw), 2024. Vol. 51, No. 2. P. 147–161. DOI: 10.4064/am2512-6-2024
21. Mehamdia A. E., Chaib Y., Bechouat T. A modified conjugate gradient method for solving unconstrained optimization with application in conditional mode regression. *Int. J. Comput. Math.*, 2025. Vol. 102, No. 9. P. 1415–1427. DOI: 10.1080/00207160.2025.2499877
22. Polak E., Ribiere G. Note sur la convergence de méthodes de directions conjuguée. *Rev. Française Informat Recherche Operationelle*, 1969. Vol. 3, No. R1. P. 35–43. (in France)
23. Polyak B. T. The conjugate gradient method in extreme problems. *USSR Comp. Math. Math. Phys.*, 1969. Vol. 9, No. 4. P. 94–112. DOI: 10.1016/0041-5553(69)90035-4
24. Powell M. J. D. Nonconvex minimization calculations and the conjugate gradient method. In: *Numerical Analysis. D.F. Griffiths (eds.)* Lecture Notes in Math., vol. 1066. Berlin, Heidelberg: Springer, 1984. Vol. 1066. P. 122–141. DOI: 10.1007/BFb0099521
25. Yao S., Wei Z., Huang H. A notes about WYL's conjugate gradient method and its applications. *Appl. Math. Comput.*, 2007. Vol. 191, No. 2. P. 381–388. DOI: 10.1016/j.amc.2007.02.094
26. Yu G., Zhao Ya., Wei Z. A descent nonlinear conjugate gradient method for large-scale unconstrained optimization. *Appl. Math. Comput.*, 2007. Vol. 187, No. 2. P. 636–643. DOI: 10.1016/j.amc.2006.08.087
27. Wei Z., Yao S., Liu L. The convergence properties of some new conjugate gradient methods. *Appl. Math. Comput.*, 2006. Vol. 183, No. 2. P. 1341–1350. DOI: 10.1016/j.amc.2006.05.150
28. Wu X. Global convergence of a modified conjugate gradient method. *J. Inequal. Appl.*, 2014. Art. no. 248. DOI: 10.1186/1029-242X-2014-248
29. Zhu Z., Zhang D., Wang S. Two modified DY conjugate gradient methods for unconstrained optimization problems. *Appl. Math. Comput.*, 2020. Vol. 373. Art. no. 125004. DOI: 10.1016/j.amc.2019.125004