Hindawi Publishing Corporation
Journal of Applied Mathematics
Volume 2012, Article ID 932980, 14 pages
doi:10.1155/2012/932980

Research Article

The Global Convergence of a New Mixed Conjugate Gradient Method for Unconstrained Optimization

Yang Yueting and Cao Mingyuan

School of Mathematics, Beihua University, Jilin 132013, China

Correspondence should be addressed to Yang Yueting, yyt2858@163.com

Received 14 June 2012; Revised 7 September 2012; Accepted 13 September 2012

Academic Editor: Hak-Keung Lam

We propose and generalize a new nonlinear conjugate gradient method for unconstrained optimization. The global convergence is proved with the Wolfe line search. Numerical experiments are reported which support the theoretical analyses and show the presented methods outperforming CGDESCENT method.

1. Introduction

This paper is concerned with conjugate gradient methods for unconstrained optimization

$$\min f(x), \quad x \in \Re^n, \tag{1.1}$$

where $f(x) : \Re^n \to \Re$ is continuously differentiable and bounded from below. Starting from an initial point x_1, a nonlinear conjugate gradient method generates sequences $\{x_k\}$ and $\{d_k\}$ by the below iteration

$$x_{k+1} = x_k + \alpha_k d_k, \quad k \geq 1, \tag{1.2}$$

where α_k is a step length which is determined by a line search and the direction d_k is generated as

$$d_k = \begin{cases} -g_k, & \text{if } k = 1, \\ -g_k + \beta_k d_{k-1}, & \text{if } k \geq 2, \end{cases} \tag{1.3}$$

where $g_k = \nabla f(x_k)$ is the gradient of $f(x)$ at x_k and β_k is a scalar.

Different conjugate gradient algorithms correspond to different choices for the scale parameter β_k. The well-known formulae of β_k are given by

$$\beta_k^{\mathrm{FR}} = \frac{\|g_k\|^2}{\|g_{k-1}\|^2},$$
$$\beta_k^{\mathrm{PR}} = \frac{g_k^{\mathrm{T}}(g_k - g_{k-1})}{\|g_{k-1}\|^2},$$
$$\beta_k^{\mathrm{DY}} = \frac{\|g_k\|^2}{d_{k-1}^{\mathrm{T}}(g_k - g_{k-1})},$$
$$\beta_k^{\mathrm{HS}} = \frac{g_k^{\mathrm{T}}(g_k - g_{k-1})}{d_{k-1}^{\mathrm{T}}(g_k - g_{k-1})}, \tag{1.4}$$

which are called Fletcher-Reeves [1] (FR), Polak-Ribière-Polyak [2] (PRP), Dai-Yuan [3] (DY), and Hestenes-Stiefel [4] (HS), respectively. Though FR and DY have strong convergence properties, they may have modest practical performance. While PRP and HS often have better computational performance, but they may not generally be convergent.

These motivate us to derive some efficient algorithms. In this paper, we focus on mixed conjugate gradient methods. These methods are combinations of different conjugate gradient methods. The aim of this paper is to propose the new methods that possess both convergence and well numerical results.

The line search in the conjugate gradient algorithms is often based on the Wolfe inexact line search

$$f(x_k + \alpha_k d_k) \le f(x_k) + \delta \alpha_k g_k^{\mathrm{T}} d_k, \tag{1.5}$$

$$g(x_k + \alpha_k d_k)^{\mathrm{T}} d_k \ge \sigma g_k^{\mathrm{T}} d_k, \tag{1.6}$$

where $0 < \delta < \sigma < 1$.

Many research on the parameter β_k have been concerned [5–7]. Such as Al-Baali [8] proved that FR global convergent with inexact line search in which $\sigma < 1/2$. Liu et al. [9] spread the results of [8] to the case of $\sigma = 1/2$. Dai and Yuan [10] gave an example when $\sigma > 1/2$, FR may produce a rise direction.

PRP is famous as the best performance of all conjugate gradient methods which is the restart method in nature. When the direction d_{k-1} is small and the factor $g_k - g_{k-1}$ in the numerator of β_k^{PRP} tends to zero, the search direction d_k is close to $-g_k$. Gilbert and Nocedal [11] proposed PRP$^+$ which is the most successful modified method, that is,

$$\beta_k^{\mathrm{PRP}^+} = \max\left\{\beta_k^{\mathrm{PRP}}, 0\right\}. \tag{1.7}$$

Dai and Yuan [12] presented DY method and proved the global convergence when the line search satisfies the Wolfe conditions. Zheng et al. [13] derived

$$\beta_k^{\text{new}} = \frac{g_k^T (g_k - d_{k-1})}{d_{k-1}^T (g_k - g_{k-1})},$$
$$\beta_k = \begin{cases} \beta_k^{\text{new}}, & \text{if } 0 < g_k^T d_{k-1} < \min\left(2, \dfrac{1}{\sigma}\right) \|g_k\|^2, \\ \beta_k^{DY}, & \text{otherwise,} \end{cases} \tag{1.8}$$

and discussed the properties of the new formulas.

HS is similar to PRP. It is equal to PRP when using the precision line search. HS satisfies the conjugate condition which is different from other methods.

Touati-Ahmed and Storey [14] gave

$$\beta_k^{TS} = \begin{cases} \beta_k^{PRP}, & \text{if } 0 \le \beta_k^{PRP} \le \beta_k^{FR}, \\ \beta_k^{FR}, & \text{otherwise.} \end{cases} \tag{1.9}$$

Dai and Chen [15] proposed

$$\beta_k = \begin{cases} \beta_k^{HS}, & \text{if } 0 < g_k^T g_{k-1} < \min\left(2, \dfrac{1}{\sigma}\right) \|g_k\|^2, \\ \beta_k^{DY}, & \text{otherwise.} \end{cases} \tag{1.10}$$

Dai and Ni [16] derived

$$\beta_k = \begin{cases} -b\beta_k^{DY}, & \text{if } \beta_k^{HS} < -b\beta_k^{DY}, \\ \beta_k^{HS}, & \text{if } -b\beta_k^{DY} \le \beta_k^{HS} \le \beta_k^{DY}, \\ \beta_k^{DY}, & \text{if } \beta_k^{HS} > \beta_k^{DY}. \end{cases} \tag{1.11}$$

Throughout the paper, $\|\cdot\|$ stands for the Euclidean norm.

Hager and Zhang (CGDESCENT) [17] proposed a conjugate gradient method with guaranteed descent which corresponds to the following choice for the update parameters: $\overline{\beta}_k^N = \max\{\beta_k^N, \eta_k\}$, where

$$\eta_k = \frac{-1}{\|d_k\| \min\{\eta, \|g_k\|\}},$$
$$\beta_k^N = \frac{1}{d_k^T y_k} \left(y_k - 2d_k \frac{\|y_k\|^2}{d_k^T y_k} \right)^T g_{k+1}. \tag{1.12}$$

Here, $\eta > 0$ is a constant. The extensive numerical tests and comparisons with other methods showed that this method has advantage in some aspects.

Zhang et al. (ZZL) [18] derived a descent modified PRP conjugate method, the direction d_k is generated by

$$d_k = \begin{cases} -g_k, & \text{if } k = 1, \\ -g_k + \beta_k^{\mathrm{PRP}} d_k - \theta_k y_{k-1}, & \text{if } k \geq 2. \end{cases} \tag{1.13}$$

The numerical results suggested that the efficiency of the MPRP method is encouraging.

Consider the above mixed techniques and the properties of the classical conjugate gradient methods, the new mixed methods will be presented. The main difference between the new methods and the existed methods are the choice of β_k and giving the generalization of the new method. Moreover, the direction generated by the new methods are descent directions of the objective function under mild conditions. In the numerical results, the method's overall performance will be given.

Firstly, we present a new formula

$$\beta_k^{\mathrm{new}} = \frac{g_k^{\mathrm{T}} g_k}{\mu \left| g_k^{\mathrm{T}} d_{k-1} \right| + d_{k-1}^{\mathrm{T}} \left(g_k - g_{k-1} \right)}. \tag{1.14}$$

The rest of the paper is organized as follows. In Section 2, we give a new mixed conjugate gradient algorithm and convergence analysis. Section 3 is devoted to a generalization of the new mixed method. In the last section, numerical results and comparisons with the CGDESCENT and ZZL methods on test problems are reported and show the advantage of the new methods.

2. A New Algorithm and Convergence Analysis

We discuss a new mixed conjugate gradient method

$$\beta_k^{*} = \begin{cases} \beta_k^{\mathrm{new}}, & \text{if } \left\| g_k \right\|^2 \geq \left| g_k^{\mathrm{T}} g_{k-1} \right|, \\ 0, & \text{otherwise}, \end{cases} \tag{2.1}$$

where $\mu \geq 1$.

Algorithm 2.1.

Step 1. Give $x_1 \in \Re^n$, $\varepsilon > 0$, $d_1 = -g_1$; $k := 1$.

Step 2. If $\|g_k\| < \varepsilon$, stop, else go to Step 3.

Step 3. Find α_k satisfying Wolfe conditions (1.5) and (1.6).

Step 4. Compute new iterative x_{k+1} by $x_{k+1} = x_k + \alpha_k d_k$.

Step 5. Compute β_k by (2.1), $d_{k+1} = -g_{k+1} + \beta_{k+1} d_k$, $k := k + 1$, go to Step 2.

In order to derive the global convergence of the algorithm, we use the following assumptions.

H 2.1 The objective function $f(x)$ is bounded in the level set as below

$$L_1 = \{x \in \Re^n \mid f(x) \leq f(x_1)\}, \tag{2.2}$$

where x_1 is the starting point.

H 2.2 $f(x)$ is continuously differentiable in a neighborhood N of L_1 and its gradient $g(x)$ is Lipschitz continuous, there exists a constant $L > 0$ such that

$$\|g(x) - g(y)\| \leq L\|x - y\|, \quad \forall x, y \in N. \tag{2.3}$$

Lemma 2.2 (see Zoutendijk condition [19]). *Suppose that H 2.1 and H 2.2 hold. If the conjugate gradient method satisfies* $g_k^T d_k < 0$*, then*

$$\sum_{k=1}^{\infty} \frac{(g_k^T d_k)^2}{\|d_k\|^2} < +\infty. \tag{2.4}$$

Theorem 2.3. *Suppose that H 2.1 and H 2.2 hold. Let* $\{x_k\}$ *and* $\{d_k\}$ *be generated by* (1.2) *and* (1.3)*, where* β_k *is computed by* (2.1)*,* α_k *satisfies Wolfe line search conditions, then* $g_k^T d_k < 0$ *holds for all* $k \geq 1$.

Proof. The conclusion can be proved by induction. When $k = 1$, we have $g_1^T d_1 = -\|g_1\|^2 < 0$. Suppose that $g_{k-1}^T d_{k-1} < 0$ hold for k. From (1.3), we have

$$\begin{aligned} g_k^T d_k &= -\|g_k\|^2 + \beta_k g_k^T d_{k-1} \\ &\leq -\|g_k\|^2 + |\beta_k| \left| g_k^T d_{k-1} \right|. \end{aligned} \tag{2.5}$$

When $\beta_k^* = 0$, it is obvious that

$$g_k^T d_k \leq -\|g_k\|^2 < 0. \tag{2.6}$$

When $\beta_k = \beta_k^{\text{new}}$, from (1.6) we have

$$\begin{aligned} d_{k-1}^T (g_k - g_{k-1}) &= g_k^T d_{k-1} - g_{k-1}^T d_{k-1} \\ &\geq \sigma g_{k-1}^T d_{k-1} - g_{k-1}^T d_{k-1} \\ &= (\sigma - 1) g_{k-1}^T d_{k-1} > 0. \end{aligned} \tag{2.7}$$

Then,

$$\begin{aligned} g_k^{\mathrm{T}} d_k &\le -\|g_k\|^2 + \frac{\|g_k\|^2}{\mu|g_k^{\mathrm{T}} d_{k-1}| + d_{k-1}^{\mathrm{T}}(g_k - g_{k-1})}|g_k^{\mathrm{T}} d_{k-1}| \\ &\le -\|g_k\|^2 + \frac{\|g_k\|^2}{\mu|g_k^{\mathrm{T}} d_{k-1}|}|g_k^{\mathrm{T}} d_{k-1}| \\ &= \left(-1 + \frac{1}{\mu}\right)\|g_k\|^2, \end{aligned} \tag{2.8}$$

from $\mu \ge 1$, then we can deduce that $g_k^{\mathrm{T}} d_k < 0$ holds for all $k \ge 1$.

Thus, the theorem is proved. □

Theorem 2.4. *Suppose that H 2.1 and H 2.2 hold. Consider Algorithm 2.1, where β_k is determined by* (2.1)*, if $g_k \ne 0$ holds for any k, then,*

$$\liminf_{k \to \infty} \|g_k\| = 0. \tag{2.9}$$

Proof. By contradiction, assume that (2.9) does not hold. Then there exists a constant $\varepsilon > 0$, such that

$$\|g_k\| > \varepsilon, \quad \forall k \ge 1. \tag{2.10}$$

From (2.1),

$$0 \le \beta_k^* \le \frac{\|g_k\|^2}{d_{k-1}^{\mathrm{T}}(g_k - g_{k-1})} = \beta_k^{\mathrm{DY}}. \tag{2.11}$$

By (1.3), if $\beta_k = \beta_k^{\mathrm{DY}}$, we derive

$$g_k^{\mathrm{T}} d_k = -\|g_k\|^2 + \frac{\|g_k\|^2}{d_{k-1}^{\mathrm{T}}(g_k - g_{k-1})} g_k^{\mathrm{T}} d_{k-1} = \beta_k^{\mathrm{DY}} g_{k-1}^{\mathrm{T}} d_{k-1}. \tag{2.12}$$

Then,

$$\beta_k^{\mathrm{DY}} = \frac{g_k^{\mathrm{T}} d_k}{g_{k-1}^{\mathrm{T}} d_{k-1}}. \tag{2.13}$$

So,

$$|\beta_k^*| \le |\beta_k^{\mathrm{DY}}| = \left|\frac{g_k^{\mathrm{T}} d_k}{g_{k-1}^{\mathrm{T}} d_{k-1}}\right|. \tag{2.14}$$

From (1.3), we have

$$d_k + g_k = \beta_k d_{k-1}. \tag{2.15}$$

By squaring the two sides of (2.15) and transferring and trimming, we get

$$\|d_k\|^2 = -\|g_k\|^2 - 2g_k^{\mathrm{T}} d_k + \beta_k^2 \|d_{k-1}\|^2. \tag{2.16}$$

Then,

$$\begin{gathered} \|d_k\|^2 \le -\|g_k\|^2 - 2g_k^{\mathrm{T}} d_k + \frac{\left(g_k^{\mathrm{T}} d_k\right)^2}{\left(g_{k-1}^{\mathrm{T}} d_{k-1}\right)^2} \|d_{k-1}\|^2, \\ \frac{\|d_k\|^2}{\left(g_k^{\mathrm{T}} d_k\right)^2} \le -\frac{2}{g_k^{\mathrm{T}} d_k} + \frac{\|d_{k-1}\|^2}{\left(g_{k-1}^{\mathrm{T}} d_{k-1}\right)^2} - \frac{\|g_k\|^2}{\left(g_k^{\mathrm{T}} d_k\right)^2} \le \frac{\|d_{k-1}\|^2}{\left(g_{k-1}^{\mathrm{T}} d_{k-1}\right)^2} + \frac{1}{\|g_k\|^2}. \end{gathered} \tag{2.17}$$

Since,

$$\begin{gathered} \frac{\|d_1\|^2}{\left(g_1^{\mathrm{T}} d_1\right)^2} = \frac{\|g_1\|^2}{\left(-g_1^{\mathrm{T}} g_1\right)^2} = \frac{1}{\|g_k\|^2}, \\ \frac{\|d_k\|^2}{\left(g_k^{\mathrm{T}} d_k\right)^2} \le \sum_{i\ge 1} \frac{1}{\|g_i\|^2}. \end{gathered} \tag{2.18}$$

From (2.10), we have

$$\sum_{i\ge 1} \frac{1}{\|g_i\|^2} \le \frac{k}{\varepsilon^2}. \tag{2.19}$$

Therefore,

$$\begin{gathered} \frac{\left(g_k^{\mathrm{T}} d_k\right)^2}{\|d_k\|^2} \ge \frac{\varepsilon^2}{k}, \\ \sum_{k\ge 1} \frac{\left(g_k^{\mathrm{T}} d_k\right)^2}{\|d_k\|^2} = +\infty. \end{gathered} \tag{2.20}$$

This is a contradiction to Lemma 2.2, the global convergence is got. □

3. Generalization of the New Method and Convergence

The generalization of the new mixed method is as follows:

$$\beta_k^{**} = \begin{cases} \lambda\beta_k^{\text{new}}, & \text{if } \|g_k\|^2 \geq |g_k^{\mathrm{T}} g_{k-1}|, \\ 0, & \text{otherwise}, \end{cases} \tag{3.1}$$

where $\beta_k^{\text{new}} = g_k^{\mathrm{T}} g_k / (\mu|g_k^{\mathrm{T}} d_{k-1}| + d_{k-1}^{\mathrm{T}}(g_k - g_{k-1})), \mu > 1 > \lambda > 0$.

Algorithm 3.1.

Step 1–*Step* 4 are the same as that of Algorithm 2.1.

Step 5. Compute β_k by (3.1), $d_{k+1} = -g_{k+1} + \beta_{k+1} d_k$, $k := k + 1$, go to Step 2.

Theorem 3.2. *Suppose that H 2.1 and H 2.2 hold. Let $\{x_k\}$ and $\{d_k\}$ be generated by* (1.2) *and* (1.3)*, where β_k is computed by* (3.1)*, α_k satisfies Wolfe line search conditions, then $g_k^{\mathrm{T}} d_k < 0$ holds for all $k \geq 1$.*

Proof. The conclusion can be proved by induction. When $k = 1$, we have $g_1^{\mathrm{T}} d_1 = -\|g_1\|^2 < 0$. Suppose that $g_{k-1}^{\mathrm{T}} d_{k-1} < 0$ holds. For k, it is obvious that if $\beta_k^{**} = 0$, then

$$g_k^{\mathrm{T}} d_k \leq -\|g_k\|^2 < 0. \tag{3.2}$$

When $\beta_k = \lambda\beta_k^{\text{new}}$, from (2.5) and (3.1), we have

$$\begin{aligned} g_k^{\mathrm{T}} d_k &\leq -\|g_k\|^2 + \frac{\lambda(g_k^{\mathrm{T}} g_k)}{\mu|g_k^{\mathrm{T}} d_{k-1}| + d_{k-1}^{\mathrm{T}}(g_k - g_{k-1})} \left|g_k^{\mathrm{T}} d_{k-1}\right| \\ &\leq -\|g_k\|^2 + \frac{\lambda\|g_k\|^2}{\mu|g_k^{\mathrm{T}} d_{k-1}|} \left|g_k^{\mathrm{T}} d_{k-1}\right| \\ &\leq -\|g_k\|^2 + \frac{\lambda}{\mu}\|g_k\|^2 \\ &= \|g_k\|^2 \left(\frac{\lambda}{\mu} - 1\right) < 0. \end{aligned} \tag{3.3}$$

To sum up, the theorem is proved. □

Theorem 3.3. *Suppose that H 2.1 and H 2.2 hold. Consider Algorithm 3.1, where β_k is determined by* (3.1)*, if $g_k \neq 0$ holds for any k, then,*

$$\liminf_{k \to \infty} \|g_k\| = 0. \tag{3.4}$$

Proof. By contradiction, assume that (3.4) does not hold. Then there exists a constant $\gamma > 0$ such that

$$\|g_k\| > \gamma, \quad \forall k \geq 1. \tag{3.5}$$

From (3.1)

$$0 \leq \beta_k^{**} \leq \frac{\lambda \|g_k\|^2}{d_{k-1}^{\mathrm{T}}(g_k - g_{k-1})} \leq \frac{\|g_k\|^2}{d_{k-1}^{\mathrm{T}}(g_k - g_{k-1})} = \beta_k^{\mathrm{DY}}. \tag{3.6}$$

By (1.3), we have

$$d_k + g_k = \beta_k d_{k-1}. \tag{3.7}$$

Then,

$$\|d_k\|^2 = -\|g_k\|^2 - 2g_k^{\mathrm{T}} d_k + \beta_k^2 \|d_{k-1}\|^2. \tag{3.8}$$

From (3.6),

$$\begin{gathered}
\|d_k\|^2 \leq -\|g_k\|^2 - 2g_k^{\mathrm{T}} d_k + \frac{(g_k^{\mathrm{T}} d_k)^2}{(g_{k-1}^{\mathrm{T}} d_{k-1})^2} \|d_{k-1}\|^2, \\
\frac{\|d_k\|^2}{(g_k^{\mathrm{T}} d_k)^2} \leq -\frac{2}{g_k^{\mathrm{T}} d_k} + \frac{\|d_{k-1}\|^2}{(g_{k-1}^{\mathrm{T}} d_{k-1})^2} - \frac{\|g_k\|^2}{(g_k^{\mathrm{T}} d_k)^2} \leq \frac{\|d_{k-1}\|^2}{(g_{k-1}^{\mathrm{T}} d_{k-1})^2} + \frac{1}{\|g_k\|^2}, \\
\frac{\|d_k\|^2}{(g_k^{\mathrm{T}} d_k)^2} \leq \sum_{i \geq 1} \frac{1}{\|g_i\|^2}.
\end{gathered} \tag{3.9}$$

By (3.5), we have

$$\begin{gathered}
\frac{(g_k^{\mathrm{T}} d_k)^2}{\|d_k\|^2} \geq \frac{\gamma^2}{k}, \\
\sum_{k \geq 1} \frac{(g_k^{\mathrm{T}} d_k)^2}{\|d_k\|^2} = +\infty.
\end{gathered} \tag{3.10}$$

This is a contradiction to Lemma 2.2, and the global convergence is proved. □

Table 1: Numerical results of the NEW1 and NEW2.

Number	Prob.	dim	k/nf/ng/t (NEW1) $\|g_k\|^2$ (NEW1)	k/nf/ng/t (NEW2) $\|g_k\|^2$ (NEW2)
1	Powell badly scaled	2	6/20/9/0.0010	9/25/11/0.0010
			0.2511	0.2709
2	Brown badly scaled	2	9/26/12/0.0010	9/26/11/0.0010
			0	9.1712e − 005
3	Trigonometric function	10	11/48/29/0.0010	13/58/37/0.0020
			0.0056	0.0061
4	Chebyquad	100	30/42/33/0.3594	61/72/65/0.4375
			0.0258	0.0207
5	Penalty function I	100	5/20/12/0.0313	5/20/12/0.0313
			5.4750e − 005	6.8047e − 005
		500	11/36/23/0.0010	10/34/20/0.0010
			1.5626e − 004	3.9363e − 004
6	Allgower	500	4/32/8/0.0156	4/32/8/0.0313
			0.0147	0.0215
7	Variable dimension	500	12/50/14/0.0625	12/50/14/0.0313
			5.9262e − 006	5.9262e − 006
		1000	16/54/16/0.0313	16/54/16/0.0313
			1.1473e − 004	1.1473e − 004
5	Penalty function I	1000	19/44/26/0.0313	22/48/30/0.0156
			0.0010	3.8348e − 004
8	Integral equation	1000	2/24/2/0.0010	2/24/2/0.0020
			0.0032	0.0058
9	Separable cubic	1000	9/12/11/0.3594	9/13/11/0.4219
			6.1814e − 004	4.0438e − 005
		3000	10/14/13/2.5469	9/13/12/2.4063
			7.2599e − 005	2.0012e − 004

4. Numerical Results

This section is devoted to test the implementation of the new methods. We compare the performance of the new methods with the CGDESCENT and ZZL methods.

All tests in this paper are implemented on a PC with 1.8 MHz Pentium IV and 256 MB SDRAM using MATLAB 6.5. If $\varepsilon = 10^{-6}$ then stop. Some classical test functions with standard starting points are selected to test the methods. These functions are widely used in the literature to test unconstrained optimization algorithms [20].

In the table, the four reported data (k/nf/ng/t) are iteration numbers/function evaluations/gradient evaluations/CPU time(s), and $\|g_k\|^2$ stands for the square of the gradient at the final iterate. When we set $\mu = 1$, $\lambda = 0.5$, $\eta = 0.01$, $\delta = 0.01$, $\sigma = 0.8$, the numerical results of the NEW1 and NEW2 are listed in Table 1 and the CGDESCENT and

Table 2: Numerical results of the CGDESCENT and NEW3.

Number	Prob.	dim	k/nf/ng/t (CGDESCENT) $\|g_k\|^2$ (CGDESCENT)	k/nf/ng/t (NEW3) $\|g_k\|^2$ (NEW3)
1	Powell badly scaled	2	6/16/7/0.0851	13/38/20/0.0313
			0.2704	0.4305
2	Brown badly scaled	2	13/33/14/0.0010	20/82/44/0.0313
			0	0.9525
3	Trigonometric function	10	12/56/33/0.0010	11/44/27/0.0313
			0.0075	0.0598
4	Chebyquad	100	64/75/67/0.4219	38/58/44/0.2344
			0.0178	0.0175
5	Penalty function I	100	5/20/12/0.0313	10/37/10/0.0313
			$5.4750e-005$	$3.4737e-004$
		500	10/34/20/0.0313	9/27/15/0.0010
			$3.8011e-005$	$2.5130e-004$
6	Allgower	500	4/32/8/0.1256	4/32/8/0.1250
			0.0259	0.0089
7	Variable dimension	500	12/50/14/0.0528	12/50/14/0.0313
			$5.9262e-006$	$5.9269e-004$
		1000	16/54/16/0.0313	16/53/16/0.0010
			$1.1473e-004$	$5.6730e-005$
5	Penalty function I	1000	22/49/31/0.0313	21/50/30/0.0625
			$3.1604e-004$	$3.4038e-004$
8	Integral equation	1000	2/24/2/0.0010	2/24/2/0.0010
			0.0028	0.0156
9	Separable cubic	1000	13/16/14/0.5469	16/20/20/0.5000
			$2.5196e-004$	$2.1274e-004$
		3000	10/14/13/2.5469	10/14/13/2.5469
			$7.6877e-005$	$2.2646e-004$

ZZL (NEW3) are listed in Table 2. When we set $\mu = 1.5$, $\lambda = 0.1$, $\eta = 10$, $\delta = 0.1$, $\sigma = 0.9$, the numerical results of the NEW 1 (NEW4) and NEW 2 (NEW5) are listed in Table 3 and the CGDESCENT (NEW6) and ZZL (NEW7) are listed in Table 4. It can be observed from Tables 1–4 that for the most of problems, the implementation of the new methods are superior to other methods from the iteration numbers, the calls of function, and gradient evaluations.

Compared with the CGDESCENT method, the new methods are effective (see Table 5).

Using the formula $N_{\text{total}} = \text{nf} + l * \text{ng}$, where l is fixed constant, let $l = 3$. By

$$\gamma_i(\text{NEW}(j)) = \frac{N_{\text{total}}(\text{NEW}(j))}{N_{\text{total}}(\text{CGDESCET})}, \tag{4.1}$$

where $j = 1, 2, \ldots, 7$; $i \in S$, S is the whole of classical problems' order. If $\gamma_i(\text{NEW}(j)) > 1$, then CGDESCENT method is regarded as better performance; if $\gamma_i(\text{NEW}(j)) = 1$, the methods have the same performances and if $\gamma_i(\text{NEW}(j)) < 1$, the new methods are performed better.

Table 3: Numerical results of the NEW4 and NEW5.

Number	Prob.	dim	$k/\mathrm{nf}/\mathrm{ng}/t$ (NEW4) $\|g_k\|^2$ (NEW4)	$k/\mathrm{nf}/\mathrm{ng}/t$ (NEW5) $\|g_k\|^2$ (NEW5)
1	Powell badly scaled	2	12/25/13/0.0010	9/21/10/0.0010
			0.3511	0.3709
2	Brown badly scaled	2	14/36/15/0.0010	9/26/10/0.0313
			$7.6024e-005$	1.1840
3	Trigonometric function	10	11/45/27/0.0010	13/51/33/0.0020
			0.0026	0.0017
4	Chebyquad		44/58/47/0.3281	72/82/73/0.3750
			0.0430	0.0251
5	Penalty function I	100	5/20/12/0.0313	5/20/12/0.0313
			$5.4750e-005$	$5.4750e-005$
		500	9/27/16/0.0156	9/27/16/0.0313
			$7.0807e-004$	$7.4979e-004$
6	Allgower	500	4/32/8/0.0010	4/32/8/0.0020
			0.0026	0.0017
7	Variable dimension	500	17/53/19/0.0313	19/53/19/0.0313
			$3.9292e-008$	$2.8061e-008$
		1000	16/54/16/0.0313	16/54/16/0.0010
			$1.1473e-004$	$1.1473e-004$
5	Penalty function I	1000	11/31/13/0.0313	11/31/13/0.0313
			0.0041	0.0036
8	Integral equation	1000	2/24/2/0.0020	2/24/2/0.0625
			0.0041	0.0936
9	Separable cubic	1000	15/17/16/0.4531	10/12/11/0.2969
			$3.8684e-004$	$7.7732e-004$
		3000	16/20/20/4.3750	14/16/15/3.3438
			$1.3230e-005$	$7.4737e-006$

We use

$$\gamma_{\text{total}}(\text{NEW}(j)) = \left(\Pi_{i\in S}\gamma_i(\text{NEW}(j))\right)^{1/|S|} \tag{4.2}$$

as a measure to compare the performance of CGDESCENT method and the new methods, where $|S|$ is the number of S. If $\gamma_{\text{total}}(\text{NEW}(j)) < 1$, then NEW$(j)$ method outperforms CGDESCENT method. The computational results are listed in Table 5.

It is obvious that $\gamma_{\text{total}}(\text{NEW}(j)) < 1$, where $j = 1, 2, 4, 5$, so we can deduce that the new methods outperform CGDESCENT method.

Table 4: Numerical results of the NEW6 and NEW7.

Number	Prob.	dim	k/nf/ng/t (NEW6) $\|g_k\|^2$ (NEW6)	k/nf/ng/t (NEW7) $\|g_k\|^2$ (NEW7)
1	Powell badly scaled	2	14/29/15/0.0313	19/54/30/0.0313
			0.4181	0.7893
2	Brown badly scaled	2	3/22/3/0.0156	41/111/69/0.0010
			$1.0000e+006$	$3.8332e+004$
3	Trigonometric function	10	2/26/2/0.0010	15/63/40/0.0010
			0.0059	0.7360
4	Chebyquad		4/14/5/0.0625	36/56/41/0.2344
			1.5078	0.0188
5	Penalty function I	100	5/20/12/0.0313	10/37/10/0.0313
			$5.4750e-005$	$3.4737e-004$
		500	9/27/15/0.0313	9/27/15/0.0010
			$7.1267e-004$	$2.5130e-004$
6	Allgower	500	4/32/8/0.0010	4/32/8/0.0938
			0.0059	0.0563
7	Variable dimension	500	17/53/19/0.0313	19/55/20/0.0010
			$2.8061e-008$	$2.4868e-006$
		1000	16/54/16/0.0313	16/53/16/0.0010
			$1.1473e-004$	$5.6730e-005$
5	Penalty function I	1000	11/31/13/0.0416	11/31/13/0.0010
			0.0010	0.0010
8	Integral equation	1000	2/24/2/0.0020	2/24/2/0.0625
			0.0041	0.0936
9	Separable cubic	1000	9/12/11/0.4063	16/18/17/0.3906
			$6.5318e-004$	$7.9504e-004$
		3000	12/16/14/3.2969	20/22/22/4.3438
			$1.7672e-005$	$5.7872e-004$

Table 5

γ_{total} (CGDESCENT)	1
γ_{total} (NEW1)	0.9220
γ_{total} (NEW2)	0.9270
γ_{total} (NEW3)	1.1347
γ_{total} (NEW4)	0.9895
γ_{total} (NEW5)	0.9415
γ_{total} (NEW6)	1.1595
γ_{total} (NEW7)	1.2164

Acknowledgments

This work is supported in part by the NNSF (11171003) of China, Key Project of Chinese Ministry of Education (no. 211039), and Natural Science Foundation of Jilin Province of China (no. 201215102).

References

[1] R. Fletcher and C. M. Reeves, “Function minimization by conjugate gradients,” *The Computer Journal*, vol. 7, pp. 149–154, 1964.

[2] B. T. Poliak, “The conjugate gradient method in extreme problems,” *USSR Computational Mathematics and Mathematical Physics*, vol. 9, pp. 94–112, 1969.

[3] Y. H. Dai and Y. Yuan, “A nonlinear conjugate gradient method with a strong global convergence property,” *SIAM Journal on Optimization*, vol. 10, no. 1, pp. 177–182, 1999.

[4] M. R. Hestenes and E. Stiefel, “Methods of conjugate gradients for solving linear systems,” *Journal of Research of the National Bureau of Standards*, vol. 49, pp. 409–436, 1952.

[5] Z. Wei, S. Yao, and L. Liu, “The convergence properties of some new conjugate gradient methods,” *Applied Mathematics and Computation*, vol. 183, no. 2, pp. 1341–1350, 2006.

[6] G. Yuan, “Modified nonlinear conjugate gradient methods with sufficient descent property for large-scale optimization problems,” *Optimization Letters*, vol. 3, no. 1, pp. 11–21, 2009.

[7] G. Yuan, X. Lu, and Z. Wei, “A conjugate gradient method with descent direction for unconstrained optimization,” *Journal of Computational and Applied Mathematics*, vol. 233, no. 2, pp. 519–530, 2009.

[8] M. Al-Baali, “Descent property and global convergence of the Fletcher-Reeves method with inexact line search,” *IMA Journal of Numerical Analysis*, vol. 5, no. 1, pp. 121–124, 1985.

[9] G. H. Liu, J. Y. Han, and H. X. Yin, “Global convergence of the Fletcher-Reeves algorithm with inexact linesearch,” *Applied Mathematics B*, vol. 10, no. 1, pp. 75–82, 1995.

[10] Y. H. Dai and Y. Yuan, “Convergence properties of the Fletcher-Reeves method,” *IMA Journal of Numerical Analysis*, vol. 16, no. 2, pp. 155–164, 1996.

[11] J. C. Gilbert and J. Nocedal, “Global convergence properties of conjugate gradient methods for optimization,” *SIAM Journal on Optimization*, vol. 2, no. 1, pp. 21–42, 1992.

[12] Y. H. Dai and Y. Yuan, “A nonlinear conjugate gradient method with a strong global convergence property,” *SIAM Journal on Optimization*, vol. 10, no. 1, pp. 177–182, 2000.

[13] X. Zheng, Z. Tian, and L. W. Song, “The global convergence of a mixed conjugate gradient method with the Wolfe line search,” *OR Transactions*, vol. 13, no. 2, pp. 18–24, 2009.

[14] D. Touati-Ahmed and C. Storey, “Efficient hybrid conjugate gradient techniques,” *Journal of Optimization Theory and Applications*, vol. 64, no. 2, pp. 379–397, 1990.

[15] Z. F. Dai and L. P. Chen, “A mixed conjugate gradient method by HS and DY,” *Journal of Computational Mathematics*, vol. 27, no. 4, pp. 429–436, 2005.

[16] Y. H. Dai and Q. Ni, *Testing Different Nonlinear Conjugate Gradient Methods*, Institute of Computational-Mathematics and Scientific/Engineering Computing, Chinese Academy of Sciences, Beijin, China, 1998.

[17] W. W. Hager and H. Zhang, “Algorithm 851: CGDESCENT, a conjugate gradient method with guaranteed descent,” *ACM Transactions on Mathematical Software*, vol. 32, no. 1, pp. 113–137, 2006.

[18] L. Zhang, W. Zhou, and D.-H. Li, “A descent modified Polak-Ribière-Polyak conjugate gradient method and its global convergence,” *IMA Journal of Numerical Analysis*, vol. 26, no. 4, pp. 629–640, 2006.

[19] G. Zoutendijk, “Nonlinear programming, computational methods,” in *Integer and Nonlinear Programming*, pp. 37–86, North-Holland, Amsterdam, The Netherlands, 1970.

[20] N. I. M. Gould, D. Orban, and P. L. Toint, “GALAHAD, a library of thread-safe Fortran 90 packages for large-scale nonlinear optimization,” *ACM Transactions on Mathematical Software*, vol. 29, no. 4, pp. 353–372, 2003.