Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2014, Article ID 394096, 9 pages
http://dx.doi.org/10.1155/2014/394096

Research Article

Two Modified Three-Term Type Conjugate Gradient Methods and Their Global Convergence for Unconstrained Optimization

Zhongbo Sun,[1,2] Yantao Tian,[1,3] and Hongyang Li[1]

[1] *College of Communication Engineering, Jilin University, Changchun 130025, China*
[2] *Department of Mathematics, College of Humanities and Sciences of Northeast Normal University, Changchun 130117, China*
[3] *Key Laboratory of Bionic Engineering of Ministry of Education, Jilin University, Changchun 130025, China*

Correspondence should be addressed to Yantao Tian; tianyt@jlu.edu.cn

Received 22 July 2014; Revised 5 November 2014; Accepted 6 November 2014; Published 23 November 2014

Academic Editor: Masoud Hajarian

Copyright © 2014 Zhongbo Sun et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Two modified three-term type conjugate gradient algorithms which satisfy both the descent condition and the Dai-Liao type conjugacy condition are presented for unconstrained optimization. The first algorithm is a modification of the Hager and Zhang type algorithm in such a way that the search direction is descent and satisfies Dai-Liao's type conjugacy condition. The second simple three-term type conjugate gradient method can generate sufficient decent directions at every iteration; moreover, this property is independent of the steplength line search. Also, the algorithms could be considered as a modification of the MBFGS method, but with different z^k. Under some mild conditions, the given methods are global convergence, which is independent of the Wolfe line search for general functions. The numerical experiments show that the proposed methods are very robust and efficient.

1. Introduction

We will consider the following optimization problem:

$$\min f(x), \quad x \in R^n, \tag{1}$$

where $f(x) : R^n \to R$ is continuously differentiable function whose gradient is $\nabla f(x)$, and $\nabla f(x)$ is denoted by $g(x)$.

Conjugate gradient method is very efficient for large-scale optimization problems. Generally, this method generates a sequence of iterations:

$$x^{k+1} = x^k + \alpha_k d^k, \quad k = 0, 1, \ldots, \tag{2}$$

where the step size α_k is obtained by carrying out some line search rules. The line search in conjugate gradient algorithms is often based on the following Wolfe conditions:

$$\begin{gathered} f\left(x^k + \alpha_k d^k\right) - f\left(x^k\right) \le \delta \alpha_k g^{kT} d^k, \\ g\left(x^k + \alpha_k d^k\right)^T d^k \ge \sigma g^{kT} d^k, \end{gathered} \tag{3}$$

where $0 < \delta \le \sigma < 1$. The direction d^k is defined by

$$d^k = \begin{cases} -g^k & \text{if } k = 0, \\ -g^k + \beta_k d^{k-1} & \text{if } k > 0, \end{cases} \tag{4}$$

where β_k is conjugate gradient parameter.

Some conjugate gradient methods include the Fletcher-Reeves method (FR), the Hestenes-Stiefel method (HS), the Polak-Ribiere-Polyak method (PRP), the conjugate descent method (CD), the Liu-Storey method (LS), and the Dai-Yuan method (DY) [2–8]. In these methods, the difference is parameter β_k; the parameters β_k of these methods are specified as follows:

$$\beta_k^{\mathrm{FR}} = \frac{\left\|g^{k+1}\right\|^2}{\left\|g^k\right\|^2}, \qquad \beta_k^{\mathrm{DY}} = \frac{\left\|g^{k+1}\right\|^2}{y^{kT} d^k},$$

$$\beta_k^{\mathrm{CD}} = -\frac{\left\|g^{k+1}\right\|^2}{g^{kT} d^k}, \qquad \beta_k^{\mathrm{HS}} = \frac{g^{k+1T} y^k}{y^{kT} d^k},$$

$$\beta_k^{\mathrm{PRP}} = \frac{g^{k+1T} y^k}{\|g^k\|^2}, \qquad \beta_k^{\mathrm{LS}} = \frac{-g^{k+1T} y^k}{g^{kT} d^k}, \tag{5}$$

where $s^k = x^{k+1} - x^k = \alpha_k d^k$, $g^k = \nabla f(x^k)$, and $y^k = g^{k+1} - g^k$. Throughout this paper, we always use $\|\cdot\|$ to mean the Euclidean norm.

2. Motivation

In this paper, the motivation will be described as follows. Firstly, we will introduce some modified three-term conjugate gradient methods. One of the first three-term conjugate gradient methods was proposed by Beale [9] as

$$d^{k+1} = -g^{k+1} + \beta_k d^k + \gamma_k d^t, \tag{6}$$

where $\beta_k = \beta_k^{\mathrm{HS}} (\beta_k^{\mathrm{FR}}, \beta_k^{\mathrm{DY}} \text{etc.})$,

$$\gamma_k = \begin{cases} 0, & \text{if } k = t+1, \\ \dfrac{g^{k+1T} y^t}{d^{tT} y^t}, & \text{if } k > t+1, \end{cases} \tag{7}$$

and d^t is a restart direction. McGuire and Wolfe [10] and Powell [11] made further study on Beale's three-term conjugate gradient method. Dai and Yuan [12] studied the general three-term conjugate gradient method:

$$d^{k+1} = -g^{k+1} + \beta_k d^k + \gamma_k d^{t(p)}, \tag{8}$$

where $t(p)$ is the number of the pth restart iteration, showing that under some mild conditions the algorithm was global convergence. As we know, Dai and Liao [13] extended the classical conjugate condition $y^{kT} d^{k+1} = 0$, suggesting the following one:

$$y^{kT} d^{k+1} = -t s^{kT} g^{k+1}, \tag{9}$$

where $t \geq 0$ is a scalar. Recently, Andrei [14, 15] developed two simple three-term conjugate gradient methods for unconstrained optimization problems. In [14], for the descent condition and Dai-Liao's type conjugacy condition, the three-term conjugate gradient algorithm was satisfied at every step. The direction d^{k+1} was computed as

$$\begin{gathered} d^{k+1} = -g^{k+1} - \delta_k s^k - \eta_k y^k, \\ \delta_k = \left(1 + \frac{\|y^k\|^2}{y^{kT} s^k}\right) \frac{s^{kT} g^{k+1}}{y^{kT} s^k} - \frac{y^{kT} g^{k+1}}{y^{kT} s^k}, \\ \eta_k = \frac{s^{kT} g^{k+1}}{y^{kT} s^k}, \end{gathered} \tag{10}$$

where δ_k and η_k were parameters. Similarly, Andrei [15] presented another project three-term conjugate gradient algorithm. The search direction of the algorithms from this class had three terms and was computed as modifications of the classical conjugate gradient algorithms. The search direction also satisfied both descent property and Dai-Liao's type conjugacy conditions. Yao and Qin [16] proposed a hybrid of DL and WYL conjugate gradient methods. The given method [17] possessed the sufficient descent condition under the Wolfe-Powell line search and is global convergence for general functions. Thereafter, they proposed a new conjugacy condition and nonlinear conjugate gradient method; the given method [17] was global convergence under the strong Wolfe-Powell line search. By modifying the HS method or Hager and Zhang method [18], the above methods with satisfying the conjugacy condition type were concluded.

In this paper, we present two modified simple three-term type conjugate gradient methods which are obtained by a modified BFGS (MBFGS) updating scheme of the inverse approximation of the Hessian of the function $f(x)$ restart as the identity matrix at every step. Firstly, in order to interpret the idea of this paper, it is necessary to introduce the MBFGS method [1]. If the objective function $f(x)$ is nonconvex, the classical Newton direction may not be a descent direction of $f(x)$ at x^k since the Hessian matrix B_k is not necessarily positive definite. To overcome this drawback, Li and Fukushima [1] generated a direction from

$$(B_k + r_k I) d^k + g^k = 0, \tag{11}$$

where I was the unit matrix and a positive constant r_k was chosen so that $B_k + r_k I$ was a positive definite matrix. In order to obtain the global convergence, the sequence $\{r_k\}$ was bounded above. For the sake of the superlinear convergence, the positive constants r_k should satisfy $r_k \to 0$ as $k \to \infty$. If the MBFGS method was quadratic convergence, the positive constants r_k should meet $r_k \leq C\|g^k\|$ for all k. In other words, the global and superlinear convergence of MBFGS method depends on the choice of $\{r_k\}$. Therefore, it was important to select $\{r_k\}$ appropriately so that it was practicable and satisfied the above conditions. How to select the optimal $\{r_k\}$ is very difficult for us. Actually, the MBFGS method was a modified quasi-Newton method for nonconvex unconstrained optimization problems. In order to better introduce our method, let us simply recall the MBFGS quasi-Newton method. The direction d^k in the MBFGS method is given by

$$d^k = -(B_k + r_k I)^{-1} g^k, \tag{12}$$

where B_k is obtained by the MBFGS formula:

$$B_{k+1} = B_k - \frac{B_k s^k s^{kT} B_k}{s^{kT} B_k s^k} + \frac{z^k z^{kT}}{z^{kT} s^k}. \tag{13}$$

From the above MBFGS formula, z^k is computed as the following two cases. On the one hand,

$$z^k = y^k + r_k s^k, \tag{14}$$

where $s^k = x^{k+1} - x^k$ and $r_k \in [0, C]$; C is a constant. On the other hand,

$$z^k = y^k + \bar{t}_k \|g^k\| s^k, \tag{15}$$

where $r_k = \bar{t}_k \|g^k\|$ and $\bar{t}_k = 1 + \max\{0, -(y^{kT} s^k / \|s^k\|^2)\}$.

The MBFGS updating of the inverse approximate of the Hessian of function $f(x)$ is

$$\begin{aligned} B_{k+1} &= B_k + \frac{\left(s^k - B_k z^k\right) s^{kT} + s^k \left(s^k - B_k z^k\right)^T}{s^{kT} z^k} \\ &\quad - \frac{z^{kT}\left(s^k - B_k z_k\right)}{\left(s^{kT} z^k\right)^2} s^k s^{kT} \\ &= \left(I - \frac{s^k z^{kT}}{s^{kT} z^k}\right) B_k \left(I - \frac{z^k s^{kT}}{s^{kT} z^k}\right) + \frac{s^k s^{kT}}{s^{kT} z^k} \\ &= B_k - \frac{s^k z^{kT} B_k + B_k z^k s^{kT}}{z^{kT} s^k} \\ &\quad + \left(1 + \frac{z^{kT} B_k z^k}{z^{kT} s^k}\right) \frac{s^k s^{kT}}{s^{kT} z^k}. \end{aligned} \tag{16}$$

In order to compass our object we take a little modification of the inverse MBFGS matrix B_k, namely, $B_k = I$; then (16) will be formulated as

$$Q_k = I - \frac{s^k z^{kT} + z^k s^{kT}}{z^{kT} s^k} + \left(1 + \frac{z^{kT} z^k}{z^{kT} s^k}\right) \frac{s^k s^{kT}}{s^{kT} z^k}, \tag{17}$$

but with different z^k. In this paper, we compute z^k as follows:

$$z^k = y^k + C \left\|g^k\right\|^r s^k + \max\left\{0, -\frac{y^{kT} s^k}{\left\|s^k\right\|^2}\right\} s^k. \tag{18}$$

From the above formula, we can see that it does not include the r_k, so we need not choose the parameters $\{r_k\}$. Furthermore, there is more information in z^k, for instance, the gradient of $f(x)$ at x^k, that is, g^k and the constants C, r. The most important property of z^k is that z^k satisfies the Dai-Liao condition; that is,

$$z^{kT} d^{k+1} = -t_k s^{kT} g^{k+1}, \tag{19}$$

where $t_k = 1 + 2(\|z^k\|^2 / z^{kT} s^k) > 0$. Moreover, z^k also satisfies the following inequality:

$$z^{kT} s^k > 0. \tag{20}$$

However, Li and Fukushima [1] proved that the z^k could satisfy the following inequality:

$$z^{kT} s^k \geq \left\|g^k\right\| \left\|s^k\right\|^2 \geq 0. \tag{21}$$

The purpose of this paper is to overcome these drawbacks. Observe that the direction d^{k+1} can be written as

$$d^{k+1} = -Q_k g^{k+1}; \tag{22}$$

therefore, the three-term type conjugate gradient algorithm is given by (2), where the direction is computed as

$$d^{k+1} = \begin{cases} -g^{k+1}, & \text{if } k = 0, \\ -g^{k+1} - \lambda_k s^k - \mu_k z^k, & \text{if } k > 0, \end{cases} \tag{23}$$

where

$$\lambda_k = \left(1 + \frac{\left\|z^k\right\|^2}{z^{kT} s^k}\right) \frac{s^{kT} g^{k+1}}{z^{kT} s^k} - \frac{z^{kT} g^{k+1}}{z^{kT} s^k}, \quad \mu_k = \frac{s^{kT} g^{k+1}}{z^{kT} s^k}. \tag{24}$$

We organize the paper as follows. In the next section, we describe the three-term type conjugate gradient method and its global convergence. In Section 4, we discuss another modified sufficient descent three-term type conjugate gradient method for unconstrained optimization problems and its global convergence with Wolfe line search. In Section 5, some numerical results are given. Some conclusions and future works will be proposed in the last section.

3. Three-Term Conjugate Gradient Method and Its Global Convergence

In this section, we will introduce the three-term type conjugate gradient method.

Algorithm 1.

Step 1 (initialization and date). Given $x^0 \in R^n, 0 < \delta \leq \sigma < 1$. Let $k := 0$.

Step 2. Test a criterion for stopping the iterations. If the test criterion $\|g^k\| \leq \epsilon$ is satisfied, then STOP! Otherwise, continue with Step 3.

Step 3. The direction d^k is computed as follows:

$$d^{k+1} = \begin{cases} -g^{k+1}, & \text{if } k = 0, \\ -g^{k+1} - \lambda_k s^k - \mu_k z^k, & \text{if } k > 0, \end{cases} \tag{25}$$

where $\lambda_k = (1 + \|z^k\|^2 / z^{kT} s^k)(s^{kT} g^{k+1} / z^{kT} s^k) - z^{kT} g^{k+1} / z^{kT} s^k$, $\mu_k = s^{kT} g^{k+1} / z^{kT} s^k$, and

$$z^k = y^k + C \left\|g^k\right\|^r s^k + \max\left\{0, -\frac{y^{kT} s^k}{\left\|s^k\right\|^2}\right\} s^k, \tag{26}$$

$C, r > 0$ are constants.

Step 4. The Wolfe line search is to find a stepsize

$$\alpha_k = \max\left\{\rho^j \mid j = 0, 1, 2, \ldots\right\} \tag{27}$$

satisfying

$$\begin{gathered} f\left(x^k + \alpha_k d^k\right) - f\left(x^k\right) \leq \delta \alpha_k g^{kT} d^k, \\ g\left(x^k + \alpha_k d^k\right)^T d^k \geq \sigma g^{kT} d^k, \end{gathered} \tag{28}$$

where $\rho \in (0, 1)$ are constants.

Step 5. Update the variables:

$$x^{k+1} = x^k + \alpha_k d^k, \tag{29}$$

and then go to Step 2.

3.1. Global Convergence. Throughout this section, we assume that

(H1) the level set $\Omega = \{x \in R^n \mid f(x) \leq f(x^0)\}$ is bounded;

(H2) in a neighborhood N of Ω the function $f(x)$ is continuously differentiable. Its gradient is Lipschitz continuous; namely, there exists a constant $L > 0$, such that

$$\|\nabla f(x) - \nabla f(y)\| \leq L \|x - y\|, \quad \forall x, y \in N. \tag{30}$$

Since $\{f(x^k)\}$ is decreasing, it is clear that the sequence $\{x^k\}$ generated by Algorithm 1 is contained in Ω. In addition, we can obtain from Algorithm 1 that there is constant κ, such that

$$\|\nabla f(x)\| \leq \kappa, \quad \forall x \in \Omega. \tag{31}$$

Lemma 2. *Let assumptions (H1) and (H2) hold; the line search satisfies the Wolfe conditions (28); then one has*

$$z^{kT} s^k > 0. \tag{32}$$

Proof. By direct computation,

$$\begin{aligned} z^{kT} s^k &= y^{kT} s^k + C \left\|g^k\right\|^r s^{kT} s^k + \max\left\{0, -\frac{y^{kT} s^k}{\left\|s^k\right\|^2}\right\} s^{kT} s^k \\ &= y^{kT} s^k + C \left\|g^k\right\|^r \left\|s^k\right\|^2 + \max\left\{0, -\frac{y^{kT} s^k}{\left\|s^k\right\|^2}\right\} \left\|s^k\right\|^2 \\ &\geq y^{kT} s^k > 0. \end{aligned} \tag{33}$$

□

Lemma 3. *Let assumptions (H1) and (H2) hold; the line search satisfies the Wolfe conditions (28); the search direction is computed by (25); then one has*

$$g^{k+1T} d^{k+1} \leq 0. \tag{34}$$

Proof. We will divide (25) into two cases as follows.

Case 1. If $k = 0$, then Lemma 3 is true.

Case 2. If $k > 0$, combined with Lemma 2, then we have

$$\begin{aligned} g^{k+1T} d^{k+1} &= g^{k+1T} \left(-g^{k+1} - \lambda_k s^k - \mu_k z^k\right) \\ &= -\left\|g^{k+1}\right\|^2 - \left(1 + \frac{\left\|z^k\right\|^2}{z^{kT} s^k}\right) \frac{\left(s^{kT} g^{k+1}\right)^2}{z^{kT} s^k} \leq 0. \end{aligned} \tag{35}$$

The proof of Lemma 3 is completed. □

Lemma 4. *Let assumptions (H1) and (H2) hold; the line search satisfies the Wolfe conditions (28); the search direction is computed by (25); then one has Dai-Liao's type conjugate condition*

$$z^{kT} d^{k+1} = -t_k s^{kT} g^{k+1}, \tag{36}$$

where $t_k > 0$.

Proof. By direct computation, we get

$$\begin{aligned} z^{kT} d^{k+1} &= z^{kT} \left(-g^{k+1} - \lambda_k s^k - \mu_k z^k\right) \\ &= -z^{kT} g^{k+1} - \left(1 + \frac{\left\|z^k\right\|^2}{z^{kT} s^k}\right) s^{kT} g^{k+1} \\ &\quad + z^{kT} g^{k+1} - \frac{s^{kT} g^{k+1}}{z^{kT} s^k} \left\|z^k\right\|^2 \\ &= -\left(1 + 2\frac{\left\|z^k\right\|^2}{z^{kT} s^k}\right) s^{kT} g^{k+1} \\ &= -t_k s^{kT} g^{k+1}, \end{aligned} \tag{37}$$

where

$$t_k = 1 + 2\frac{\left\|z^k\right\|^2}{z^{kT} s^k} > 0, \tag{38}$$

since $z^{kT} s^k > 0$. □

Remark 5. Observe that the direction d^{k+1} satisfies descent property. Besides, the direction d^{k+1} satisfies Dai-Liao's type conjugacy condition, where $t_k > 0$ at every iteration.

Theorem 6. *Let assumptions (H1) and (H2) hold; the line search satisfies the Wolfe conditions (28); the search direction is computed by (25); then one has*

$$\left\|z^k\right\| \leq \nu_1 \left\|d^k\right\|, \qquad \left\|\lambda_k\right\| \leq \nu_2, \qquad \left\|\mu_k\right\| \leq \nu_3. \tag{39}$$

Proof. By direct computation, we have

$$\begin{aligned} \left\|z^k\right\| &\leq \left\|y^k\right\| + C \left\|g^k\right\|^r \left\|s^k\right\| \\ &\quad + \max\left\{0, -\frac{y^{kT} s^k}{\left\|s^k\right\|^2}\right\} \left\|s^k\right\| \\ &\leq L \left\|d^k\right\| + C \left\|g^k\right\|^r \left\|d^k\right\| + L \left\|d^k\right\| \\ &= \left(2L + C\kappa^r\right) \left\|d^k\right\| = \nu_1 \left\|d^k\right\|, \end{aligned} \tag{40}$$

where $\nu_1 = 2L + C\kappa^r$. Consider

$$\begin{aligned} \left\|\lambda_k\right\| &\leq \left(1 + \frac{\left\|z_k\right\|^2}{z^{kT} s^k}\right) \left|\frac{s^{kT} g^{k+1}}{z^{kT} s^k}\right| + \left|\frac{z^{kT} g^{k+1}}{z^{kT} s^k}\right| \\ &\leq \left(1 + \frac{\left\|z_k\right\|^2}{y^{kT} s^k}\right) \left|\frac{s^{kT} g^{k+1}}{y^{kT} s^k}\right| + \left|\frac{z^{kT} g^{k+1}}{y^{kT} s^k}\right| \\ &\leq \left[1 + \frac{\left\|z_k\right\|^2}{(1-\sigma) \left\|d^k\right\|^2}\right] \frac{1}{(1-\sigma)} + \frac{\nu_1}{(1-\sigma)} \\ &\leq \left[1 + \frac{\kappa^2}{(1-\sigma)}\right] \frac{1}{(1-\sigma)} + \frac{\nu_1}{(1-\sigma)} = \nu_2, \end{aligned} \tag{41}$$

where $\nu_2 = [1 + \kappa^2/(1-\sigma)](1/(1-\sigma)) + \nu_1/(1-\sigma)$. Consider

$$\left\|\mu_k\right\| = \left|\frac{s^{kT}g^{k+1}}{z^{kT}s^k}\right| \leq \left|\frac{s^{kT}g^{k+1}}{y^{kT}s^k}\right| \leq \frac{1}{1-\sigma} = \nu_3, \tag{42}$$

where $\nu_3 = (1/(1-\sigma))$. The proof is completed. □

Theorem 7. *Let assumptions (H1) and (H2) hold; the line search satisfies the Wolfe conditions (28); the search direction is computed by (25); then one has*

$$\left\|d^{k+1}\right\| \leq \omega \left\|g^{k+1}\right\|. \tag{43}$$

Proof. According to Algorithm 1, we will divide (28) into two cases as follows.

Case 1. If $k = 0$, we get $d^{k+1} = -g^{k+1}$, and then

$$\left\|d^{k+1}\right\| = \left\|g^{k+1}\right\|, \tag{44}$$

where $\omega = 1$.

Case 2. If $k > 0$, we have

$$d^{k+1} = -g^{k+1} - \lambda_k s^k - \mu_k z^k, \tag{45}$$

and then

$$\begin{aligned}\left\|d^{k+1}\right\| &\leq \left\|g^{k+1}\right\| + \left\|\lambda_k\right\| \left\|s^k\right\| + \left\|\mu_k\right\| \left\|z^k\right\| \\ &\leq \left\|g^{k+1}\right\| + \left\|\left(1 + \frac{\left\|z^k\right\|^2}{z^{kT}s^k}\right)\frac{s^{kT}g^{k+1}}{z^{kT}s^k}s^k\right\| \\ &\quad + \left\|\frac{z^{kT}g^{k+1}}{z^{kT}s^k}s^k\right\| + \left\|\frac{s^{kT}g^{k+1}}{z^{kT}s^k}z^k\right\| \leq \left\|g^{k+1}\right\| \\ &\quad + \left(1 + \frac{\left\|z^k\right\|^2}{z^{kT}s^k}\right)\frac{\left\|s^k\right\|\left\|g^{k+1}\right\|}{z^{kT}s^k}\left\|s^k\right\| \\ &\quad + \frac{\left\|z^k\right\|\left\|g^{k+1}\right\|}{z^{kT}s^k}\left\|s^k\right\| + \frac{\left\|s^k\right\|\left\|g^{k+1}\right\|}{z^{kT}s^k}\left\|z^k\right\| \\ &= \omega\left\|g^{k+1}\right\|.\end{aligned} \tag{46}$$

The proof is true. □

Theorem 8. *Let assumption (H1) hold and let $\{x^k\}$ and $\{d^k\}$ be generated by the three-term conjugate gradient method. If stepsize α_k is obtained by the Wolfe line search, then one has*

$$\lim_{k\to\infty}\inf\left\|g^k\right\| = 0. \tag{47}$$

Proof. By the use of Theorem 7 and the same argument as Theorem 4.1 in [19], we will omit the proof here. □

4. Another Modified Three-Term Type Conjugate Gradient Method and Its Global Convergence

Recently, Zhang et al. proposed a sufficient descent modified PRP conjugate gradient method with three terms [20] as

$$d^{k+1} = -g^{k+1} + \frac{g^{k+1T}y^k}{\left\|g^k\right\|^2}d^k - \frac{g^{k+1T}d^k}{\left\|g^k\right\|^2}y^k \tag{48}$$

and a sufficient descent modified HS conjugate gradient method with three terms [21] as

$$d^{k+1} = -g^{k+1} + \frac{g^{k+1T}y^k}{s^{kT}y^k}s^k - \frac{g^{k+1T}s^k}{s^{kT}y^k}y^k. \tag{49}$$

A property of these methods is that they produce sufficient direction; that is,

$$g^{kT}d^k = -\left\|g^k\right\|^2. \tag{50}$$

In the same context, Sun et al. proposed another sufficient conjugate gradient [22, 23] as

$$d^k = \begin{cases}-g^k, & \text{if } k = 0,\\ -g^k + \beta_{k1}d^{k-1} - \vartheta_k y^k, & \text{if } k > 0,\end{cases} \tag{51}$$

where $\beta_{k1} = g^{kT}(g^k - g^{k-1})/((1-\mu)\|g^{k-1}\|^2 - \mu g^{k-1T}d^{k-1})$ and

$$d^k = \begin{cases}-g^k, & \text{if } k = 0,\\ -\theta_k g^k + \beta_k^{\mathrm{VDY}}d^{k-1} & \text{if } k > 0,\end{cases} \tag{52}$$

where

$$\theta_k = 1 + \frac{\mu_1 g^{kT}d^{k-1}}{\mu_2\left|g^{kT}d^{k-1}\right| + \mu_3 d^{k-1T}y^{k-1}}. \tag{53}$$

Similar to Zhang and Sun's methods, in order to obtain the sufficient descent property, we will propose a modified three-term type conjugate gradient method; that is,

$$d^{k+1} = \begin{cases}-g^{k+1}, & \text{if } k = 0,\\ -\xi_k g^{k+1} - \lambda_k s^k - \mu_k z^k, & \text{if } k > 0,\end{cases} \tag{54}$$

where

$$\xi_k = \frac{\lambda_k g^{k+1T}s^k + \mu_k g^{k+1T}z^k - \left\|g^{k+1}\right\|^2}{\left\|g^{k+1}\right\|^2}. \tag{55}$$

Algorithm 9 (a sufficient decent three-term conjugate gradient method).

Step 1. Step 1 is initialization and date.

Step 2 (termination condition and computation d^k). The direction d^k is computed by (54).

Step 3. The Wolfe line search is to find a stepsize α_k.

Step 4. Update the variables, $x^{k+1} = x^k + \alpha_k d^k$, and then go to Step 2.

Lemma 10. *Suppose that x^0 is a starting point for which assumptions hold. Let $\{x^k\}$ be generated by Algorithm 9; then one has*

$$g^{k+1T}d^{k+1} = -\left\|g^{k+1}\right\|^2 < 0. \tag{56}$$

Proof. According to Algorithm 9, we will divide (54) into two cases as follows.

Case 1. If $k = 0$, then Lemma 10 is true.

Case 2. If $k > 0$, we get

$$\begin{aligned} g^{k+1T}d^{k+1} &= g^{k+1T}\left(-\xi_k g^{k+1} - \lambda_k s^k - \mu_k z^k\right) \\ &= -\xi_k \left\|g^{k+1}\right\|^2 - \left(1 + \frac{\left\|z^k\right\|^2}{z^{kT}s^k}\right)\frac{\left(s^{kT}g^{k+1}\right)^2}{z^{kT}s^k} \\ &\quad - \frac{\left(s^{kT}g^{k+1}\right)^2}{z^{kT}s^k} = -\left\|g^{k+1}\right\|^2 < 0. \end{aligned} \tag{57}$$

Lemma 10 is true. The sufficient descent property is independent of the line search. □

Lemma 11. *Suppose that x^0 is a starting point for which assumptions hold. Let $\{x^k\}$ be generated by Algorithm 9. In addition, there are constants κ_1, κ_2, such that*

$$\kappa_1 \le \left\|g^k\right\| \le \kappa_2, \quad \forall x \in \Omega; \tag{58}$$

then one has

$$\left|\xi_k\right| \le \psi. \tag{59}$$

Proof. By direct computation, we get

$$\begin{aligned} \xi_k &= \left|\frac{\lambda_k g^{k+1T}s^k + \mu_k g^{k+1T}z^k - \left\|g^{k+1}\right\|^2}{\left\|g^{k+1}\right\|^2}\right| \\ &\le 1 + \frac{\left|\lambda_k\right|\left\|g^{k+1}\right\|\left\|s^k\right\| + \left|\mu_k\right|\left\|g^{k+1}\right\|\left\|z^k\right\|}{\left\|g^{k+1}\right\|^2} \\ &\le 1 + \frac{\nu_2\kappa_2\left\|d^k\right\| + \nu_3\kappa_2 L\left\|d^k\right\|}{\left\|\kappa_1\right\|^2} \\ &\le 1 + \frac{\nu_2\kappa_2^2\omega + \nu_3\kappa_2^2 L\omega}{\left\|\kappa_1\right\|^2} = \psi, \end{aligned} \tag{60}$$

where

$$\psi = 1 + \frac{\nu_2\kappa_2^2\omega + \nu_3\kappa_2^2 L\omega}{\left\|\kappa_1\right\|^2}. \tag{61}$$

□

Theorem 12. *Let assumption (H1) hold and let $\{x^k\}$ and $\{d^k\}$ be generated by the three-term conjugate gradient Algorithm 9. If stepsize α_k is obtained by the Wolfe line search, then one has*

$$\lim_{k\to\infty}\inf\left\|g^k\right\| = 0. \tag{62}$$

Proof. By the use of Lemmas 10 and 11 and the same argument as Theorem 4.1 in [19], we will omit the proof here. □

5. Numerical Experiments

Now, let us report some numerical results attained by our sufficient descent conjugate gradient methods. We compare the performance of Algorithms 1 and 9 with MBFGS method. The algorithm is implemented by MATLAB 7.0 code in double precision arithmetic. The tests are performed on a PC computer with CPU Pentium 4, 2.40 GHz, and Windows XP operation system.

On the one hand, the type of objective function and the character of the problems being tested are listed in Tables 1, 2, and 3. In the experiments, for easily comparing with other codes, we use the gradient errors to measure the quality of the solutions; that is, we force the iteration to stop when

$$\left\|g^k\right\| \le 10^{-6}, \tag{63}$$

where g^k is the gradient of objective function. n represents the number of dimensions. m represents the number of functions. Nf represents the number of function $f(x)$ evaluations. Ng represents the number of gradient $g(x)$ evaluations. S represents the actual CPU-time costed in procedure operation. f^* represents the problems approximate solutions which are allowed in error range region. P represents the problems in [24]. First, we compare Algorithm 1, Algorithm 9, and MBFGS method and the numerical test reports and the results of comparison are listed in Tables 1–3. We choose some test problems as our numerical examples and numerical results can be seen in Tables 1–3. The test problems with the given initial points can be found at http://camo.ici.ro/neculai/ansoft.htm which were collected by Neculai Andrei. We can see that the problems of [22] are solved by our method. Table 1 shows the numerical results of the three-term conjugate gradient Algorithm 1. In Table 2, by adopting Algorithm 9, these problems have better solutions. Table 3 shows the numerical results of the MBFGS method. In Table 2, the CPU-time is less than 130 seconds and lots of the problems are less than 70 seconds. In Table 1, the iterations of the Penalty function II and Power singular are more than 110, but, in Table 2, the Penalty function II and Power singular can be solved in less than 86 iterations and the time is less than 72 seconds. Table 3 shows the performance of the MBFGS method relative to CPU-time, the number of iterations, the number of function evaluations, and the number of gradient evaluations, respectively. From Table 1 to Table 3, as can be easily seen, Table 3 is better than Table 1 but worse than Table 2 with respect to the number of iterations and CPU-time. Tables 1 and 2 show that the three-term type conjugate gradient methods also have better performance with respect to the number of iterations and gradient evaluations. From Tables 1, 2, and 3, a conclusion is made that Algorithm 9 is better than Algorithm 1 and MBFGS method; that is, the sufficient descent direction is most important for the unconstrained optimization.

Table 1: The numerical results of Algorithm 1.

P	n	m	Nf	Ng	S	f^*
Rosenbrock	2	2	184	183	29.6	$5.38e-006$
Freudenstein and Roth	2	2	77	78	96.2	$5.72e-006$
Brown badly scaled	2	3	106	108	453.1	$1.48e-006$
Beale	2	3	98	100	62.8	$7.31e-006$
Brown and Dennis	4	20	77	79	100.1	85822.2
Trigonometrix	100	100	10	12	45.3	$5.72e-006$
Trigonometrix	1000	1000	35	38	55.4	$5.72e-006$
Trigonometrix	5000	5000	74	78	103.3	$5.72e-006$
Broyden tridiagonal	100	100	56	56	89.2	$9.54e-006$
Broyden tridiagonal	1000	1000	79	80	145.5	$9.54e-006$
Broyden tridiagonal	5000	5000	82	86	170.2	$9.54e-006$
Broyden banded	100	100	11	11	78.2	$9.94e-006$
Broyden banded	500	500	46	46	90.3	$9.94e-006$
Power singular	4	4	113	116	77	$9.38e-006$
Brown almost linear	100	100	25	28	46.5	$9.85e-006$
Brown almost linear	1000	1000	55	58	79.2	$9.85e-006$
Watson	31	31	17	17	99.5	$8.98e-006$
Discrete boundary value	100	100	42	42	36.2	$9.54e-006$
Penalty function I	100	101	135	136	89.6	$7.11e-007$
Penalty function II	100	200	56	56	70.2	$5.69e-007$
Penalty function II	500	1000	102	105	144.6	$5.69e-007$
Penalty function II	1000	2000	256	256	403.4	$5.69e-007$

Table 2: The numerical results of Algorithm 9.

P	n	m	Nf	Ng	S	f^*
Rosenbrock	2	2	18	18	10	$5.38e-006$
Freudenstein and Roth	2	2	47	48	28.3	$5.72e-006$
Brown badly scaled	2	3	106	108	301.4	$1.48e-006$
Beale	2	3	40	40	31	$7.31e-006$
Brown and Dennis	4	20	23	23	36.4	85822.2
Trigonometrix	100	100	5	5	25	$5.72e-006$
Trigonometrix	1000	1000	20	21	40.3	$5.72e-006$
Trigonometrix	5000	5000	35	35	53.4	$5.72e-006$
Power singular	4	4	45	45	63.2	$9.38e-006$
Broyden tridiagonal	100	100	40	40	68	$9.54e-006$
Broyden tridiagonal	1000	1000	54	54	85.4	$9.54e-006$
Broyden tridiagonal	5000	5000	72	75	130.1	$9.54e-006$
Broyden banded	100	100	7	7	60.7	$9.94e-006$
Broyden banded	500	500	31	35	84.5	$9.94e-006$
Brown almost linear	100	100	25	28	33.5	$9.85e-006$
Brown almost linear	1000	1000	50	50	65.8	$9.85e-006$
Watson	31	31	14	17	75.6	$8.98e-006$
Discrete boundary value	100	100	42	42	30.4	$9.54e-006$
Penalty function I	100	101	55	56	69.1	$7.11e-007$
Penalty function II	100	200	41	41	64.3	$5.69e-007$
Penalty function II	500	1000	56	56	68	$5.69e-007$
Penalty function II	1000	2000	86	86	70.1	$5.69e-007$

Table 3: The numerical results of the MBFGS method [1].

P	n	m	Nf	Ng	S	f^*
Rosenbrock	2	2	30	30	32	6.14e − 006
Freudenstein and Roth	2	2	45	46	45.8	5.13e − 006
Brown badly scaled	2	3	120	113	317.6	0.48e − 007
Beale	2	3	56	56	45	5.33e − 006
Brown and Dennis	4	20	23	34	39.7	85856.2
Trigonometrix	100	100	23	20	53	5.72e − 006
Trigonometrix	1000	1000	34	38	121	5.72e − 006
Trigonometrix	5000	5000	45	45	135	5.72e − 006
Broyden tridiagonal	100	100	60	69	77	9.24e − 006
Broyden tridiagonal	1000	1000	90	91	186.4	9.06e − 006
Broyden tridiagonal	5000	5000	122	123	140.7	0.54e − 007
Broyden banded	100	100	22	33	100	0.91e − 007
Broyden banded	500	500	22	22	90	0.94e − 007
Power singular	4	4	56	85	78	0.34e − 008
Brown almost linear	100	100	56	55	143.3	0.87e − 006
Brown almost linear	1000	1000	50	58	128.8	0.85e − 007
Watson	31	31	22	22	86.5	8.98e − 006
Discrete boundary value	100	100	55	55	42.3	9.54e − 006
Penalty function I	100	101	60	60	88	7.11e − 007
Penalty function II	100	200	157	159	185.6	0.69e − 008
Penalty function II	500	1000	256	257	200.4	0.19e − 008
Penalty function II	1000	2000	287	287	210.4	0.69e − 008

On the other hand, some of the test problems are from the CUTE collection established by Bongartz, Conn, Gould, and Toint. In Figure 1, we adopt the performance profiles proposed by Dolan and Moré [25] to compare the CPU-time of Algorithm 1, Algorithm 9, MBFGS method, Dai's method, and MPRP method. That is, for each method, we plot the fraction **P** of problems for which the method is within a factor t of the best time. The left side of the figure gives the percentage of the test problems for which method is the fastest. The right side gives the percentage of the test problems that are successfully solved by each of the methods. The top curve is the method that solved most of the problems in a time that is within a factor t of the best time. From Figure 1, we can see that Algorithm 1 method and Algorithm 9 method perform better than the MBFGS method and Dai's method and MPRP method in [1, 19, 20]. Hence, the proposed methods not only possess better global convergence but also are superior to the three-term type conjugate gradient methods [1, 19, 20] in the numerical performance.

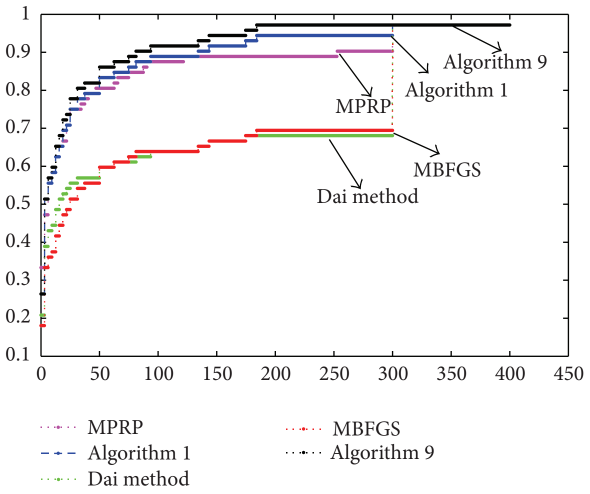

Figure 1: Performance based on CPU-time.

6. Conclusions

In this paper, on the one hand, we improve a three-term type conjugate gradient method which is obtained by a MBFGS. On the other hand, we show another sufficient decent three-term type conjugate gradient method. In addition, under appropriate conditions, we indicate that the two methods are global convergence. Finally, some numerical experiments manifest the efficiency of the proposed methods.

Certainly, we should further investigate more useful, powerful, and practical algorithms for solving large-scale unconstrained optimization problems, for instance, the hybrid conjugate gradient-GA and conjugate gradient-PSO methods and so on.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

The authors wish to thank the reviewers for their constructive and pertinent suggestions for improving the presentation of the work. The authors would like to thank Dr. Zhifeng Dai and Boshi Tian and Xiaoliang Dong for several helpful discussions regarding numerical experiments. This work is supported by the National High Technology Research and Development Program of China (863 Program), Grant no. 2006AA04Z251, the National Natural Fund Project, Grant no. 60974067, and the Funds of Jilin Province Science and Technology, Grant nos. 2013577, 2013267, and 2013287.

References

[1] D.-H. Li and M. Fukushima, "A modified BFGS method and its global convergence in nonconvex minimization," *Journal of Computational and Applied Mathematics*, vol. 129, no. 1-2, pp. 15–35, 2001.

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

[3] 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.

[4] M. R. Polak and G. Ribiere, "Note surla convergence des methodes de directions conjuguees," *Revue Française d'Automatique, Informatique, Recherche Opérationnelle*, vol. 16, pp. 35–43, 1969.

[5] B. T. Polyak, "The conjugate gradient method in extremal problems," *USSR Computational Mathematics and Mathematical Physics*, vol. 9, no. 4, pp. 94–112, 1969.

[6] R. Fletcher, *Unconstained Optimization: Practical Methods of Optimization*, vol. 1, John Wiley & Sons, New York, NY, USA, 1987.

[7] Y. Liu and C. Storey, "Efficient generalized conjugate gradient algorithms, Part 1," *Journal of Optimization Theory and Applications*, vol. 69, no. 1, pp. 129–137, 1991.

[8] Y. H. Dai and Y. Yuan, "An efficient hybrid conjugate gradient method for unconstrained optimization," *Annals of Operations Research*, vol. 103, pp. 33–47, 2001.

[9] E. M. L. Beale, "A derivative of conjugate gradients," in *Numerical Methods for Nolinear Optimization*, F. A. Lootsma, Ed., pp. 39–43, Academic Press, London, UK, 1972.

[10] M. F. McGuire and P. Wolfe, "Evaluating a restart procedure for conjugate gradients," Tech. Rep. RC-4382, IBM Research Center, Yorktown Heights, NY, USA, 1973.

[11] M. J. D. Powell, "Nonconvex minimization calculations and the conjugate gradient method," in *Numerical Analysis*, vol. 1066 of *Lecture Notes in Mathematics*, pp. 122–141, Springer, Berlin, Germany, 1984.

[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, 1999.

[13] Y.-H. Dai and L.-Z. Liao, "New conjugacy conditions and related nonlinear conjugate gradient methods," *Applied Mathematics and Optimization*, vol. 43, no. 1, pp. 87–101, 2001.

[14] N. Andrei, "A simple three-term conjugate gradient algorithm for unconstrained optimization," *Journal of Computational and Applied Mathematics*, vol. 241, pp. 19–29, 2013.

[15] N. Andrei, "On three-term conjugate gradient algorithms for unconstrained optimization," *Applied Mathematics and Computation*, vol. 219, no. 11, pp. 6316–6327, 2013.

[16] S. Yao and B. Qin, "A hybrid of DL and WYL nonlinear conjugate gradient methods," *Abstract and Applied Analysis*, vol. 2014, Article ID 279891, 9 pages, 2014.

[17] S. Yao, X. Lu, and B. Qin, "A modified conjugacy condition and related nonlinear conjugate gradient method," *Mathematical Problems in Engineering*, vol. 2014, Article ID 710376, 9 pages, 2014.

[18] W. W. Hager and H. Zhang, "A new conjugate gradient method with guaranteed descent and an efficient line search," *SIAM Journal on Optimization*, vol. 16, no. 1, pp. 170–192, 2005.

[19] Y. H. Dai, "On the nonmonotone line search," *Journal of Optimization Theory and Applications*, vol. 112, no. 2, pp. 315–330, 2002.

[20] 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.

[21] L. Zhang, W. Zhou, and D. Li, "Global convergence of a modified Fletcher-Reeves conjugate gradient method with Armijotype line search," *Numerische Mathematik*, vol. 104, no. 4, pp. 561–572, 2006.

[22] Z. Sun, T. Zhu, S. Weng, and H. Gao, "A new family sufficient descent conjugate gradient methods for unconstrained optimization," in *Proceedings of the 23rd Chinese Control Conference (CCC '12)*, pp. 2532–2536, July 2012.

[23] Z. B. Sun, T. Zhu, and H. Gao, "A sufficient descent hybrid conjugate gradient method and its global convergence for unconstrained optimization," in *Proceedings of the 24th Chinese Control and Decision Conference (CCDC '12)*, pp. 735–739, May 2012.

[24] J. J. Moré, B. S. Garbow, and K. E. Hillstrom, "Testing unconstrained optimization software," *ACM Transactions on Mathematical Software*, vol. 7, no. 1, pp. 17–41, 1981.

[25] E. D. Dolan and J. J. Moré, "Benchmarking optimization software with performance profiles," *Mathematical Programming*, vol. 91, no. 2, pp. 201–213, 2002.

Advances in Operations Research
Advances in Decision Sciences
Journal of Applied Mathematics
Algebra
Journal of Probability and Statistics
The Scientific World Journal
Hindawi
Submit your manuscripts at http://www.hindawi.com
International Journal of Differential Equations
International Journal of Combinatorics
Advances in Mathematical Physics
Journal of Complex Analysis
Journal of Mathematics
Mathematical Problems in Engineering
Abstract and Applied Analysis
Discrete Dynamics in Nature and Society
International Journal of Mathematics and Mathematical Sciences
Journal of Discrete Mathematics
Journal of Function Spaces
International Journal of Stochastic Analysis
Journal of Optimization