

RESEARCH

Open Access



Least-squares-based three-term conjugate gradient methods

Chunming Tang^{1*} , Shuangyu Li¹ and Zengru Cui¹

*Correspondence:

cmtang@gxu.edu.cn

¹College of Mathematics and Information Science, Guangxi University, Nanning, P.R. China

Abstract

In this paper, we first propose a new three-term conjugate gradient (CG) method, which is based on the least-squares technique, to determine the CG parameter, named LSTT. And then, we present two improved variants of the LSTT CG method, aiming to obtain the global convergence property for general nonlinear functions. The least-squares technique used here well combines the advantages of two existing efficient CG methods. The search directions produced by the proposed three methods are sufficient descent directions independent of any line search procedure. Moreover, with the Wolfe–Powell line search, LSTT is proved to be globally convergent for uniformly convex functions, and the two improved variants are globally convergent for general nonlinear functions. Preliminary numerical results are reported to illustrate that our methods are efficient and have advantages over two famous three-term CG methods.

MSC: 90C30; 65K05; 49M37

Keywords: Three-term conjugate gradient method; Least-squares technique; Sufficient descent property; Wolfe–Powell line search; Global convergence

1 Introduction

Consider the following unconstrained optimization problem:

$$\min_{x \in \mathbb{R}^n} f(x),$$

where $f : \mathbb{R}^n \rightarrow \mathbb{R}$ is a continuously differentiable function whose gradient function is denoted by $g(x)$.

Conjugate gradient (CG) methods are known to be among the most efficient methods for unconstrained optimization due to their advantages of simple structure, low storage, and nice numerical behavior. CG methods have been widely used to solve practical problems, especially large-scale problems such as image recovery [1], condensed matter physics [2], environmental science [3], and unit commitment problems [4–6].

For the current iteration point x_k , the CG methods yield the new iterate x_{k+1} by the formula

$$x_{k+1} = x_k + \alpha_k d_k, \quad k = 0, 1, \dots,$$

© The Author(s) 2020. This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

where α_k is the stepsize determined by a certain line search and d_k is the so-called search direction in the form of

$$d_k = \begin{cases} -g_k, & k = 0, \\ -g_k + \beta_k d_{k-1}, & k \geq 1, \end{cases}$$

in which β_k is a parameter. Different choices of β_k correspond to different CG methods. Some classical and famous formulas of the CG methods parameter β_k are:

$$\beta_k^{\text{HS}} = \frac{g_k^T y_{k-1}}{d_{k-1}^T y_{k-1}}, \quad \text{Hestenes and Stiefel (HS) [7];}$$

$$\beta_k^{\text{FR}} = \frac{\|g_k\|^2}{\|g_{k-1}\|^2}, \quad \text{Fletcher and Reeves (FR) [8];}$$

$$\beta_k^{\text{PRP}} = \frac{g_k^T y_{k-1}}{\|g_{k-1}\|^2}, \quad \text{Polak, Ribière, and Polyak (PRP) [9, 10];}$$

$$\beta_k^{\text{DY}} = \frac{\|g_k\|^2}{d_{k-1}^T y_{k-1}}, \quad \text{Dai and Yuan (DY) [11],}$$

where $g_k = g(x_k)$, $y_{k-1} = g_k - g_{k-1}$, and $\|\cdot\|$ denotes the Euclidean norm.

Here are two commonly used line searches for choosing the stepsize α_k .

- *The Wolfe–Powell line search*: the stepsize α_k satisfies the following two relations:

$$f(x_k + \alpha_k d_k) - f(x_k) \leq \delta \alpha_k g_k^T d_k \tag{1}$$

and

$$g(x_k + \alpha_k d_k)^T d_k \geq \sigma g_k^T d_k, \tag{2}$$

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

- *The strong Wolfe–Powell line search*: the stepsize α_k satisfies both (1) and the following relation:

$$|g(x_k + \alpha_k d_k)^T d_k| \leq \sigma |g_k^T d_k|.$$

In recent years, based on the above classical formulas and line searches, many variations of CG methods have been proposed, including spectral CG methods [12, 13], hybrid CG methods [14, 15], and three-term CG methods [16, 17]. Among them, the three-term CG methods seem to attract more attention, and a great deal of efforts has been devoted to developing this kind of methods, see, e.g., [18–23]. In particular, by combining the PRP method [9, 10] with the BFGS quasi-Newton method [24], Zhang et al. [22] presented a three-term PRP CG method (TTPRP). Their motivation is that the PRP method has good numerical performance but is generally not a descent method when the Armijo-type line search is executed. The direction of TTPRP is given by

$$d_k^{\text{TTPRP}} = \begin{cases} -g_k, & \text{if } k = 0, \\ -g_k + \beta_k^{\text{PRP}} d_{k-1} - \theta_k^{(1)} y_{k-1}, & \text{if } k \geq 1, \end{cases}$$

where

$$\theta_k^{(1)} = \frac{g_k^T d_{k-1}}{\|g_{k-1}\|^2}, \tag{3}$$

which is always a descent direction (independent of line searches) for the objective function.

In the same way, Zhang et al. [25] presented a three-term FR CG method (TTFR) whose direction is in the form of

$$d_k^{TTFR} = \begin{cases} -g_k, & \text{if } k = 0, \\ -g_k + \beta_k^{FR} d_{k-1} - \theta_k^{(1)} g_k, & \text{if } k \geq 1, \end{cases}$$

where $\theta_k^{(1)}$ is given by (3). Later, Zhang et al. [23] proposed a three-term HS CG method (TTHS) whose direction is defined by

$$d_k^{TTHS} = \begin{cases} -g_k, & \text{if } k = 0, \\ -g_k + \beta_k^{HS} d_{k-1} - \theta_k^{(2)} y_{k-1}, & \text{if } k \geq 1, \end{cases} \tag{4}$$

where

$$\theta_k^{(2)} = \frac{g_k^T d_{k-1}}{d_{k-1}^T y_{k-1}}.$$

The above approaches [22, 23, 25] have a common advantage that the relation $d_k^T g_k = -\|g_k\|^2$ holds. This means that they always generate descent directions without the help of line searches. Moreover, they can all achieve global convergence under suitable line searches.

Before putting forward the idea of our new three-term CG methods, we first briefly review a hybrid CG method (HCG) proposed by Babaie-Kafaki and Ghanbari [26], in which the search direction is in the form of

$$d_k^{HCG} = \begin{cases} -g_k, & \text{if } k = 0, \\ -g_k + \beta_k^{HCG} d_{k-1}, & \text{if } k \geq 1, \end{cases}$$

where the parameter is given by a convex combination of FR and PRP formulas

$$\beta_k^{HCG} = (1 - \theta_k) \beta_k^{PRP} + \theta_k \beta_k^{FR}, \quad \text{with } \theta_k \in [0, 1].$$

It is obvious that the choice of θ_k is very critical for the practical performance of the HCG method. By taking into account that the TTHS method has good theoretical property and numerical performance, Babaie-Kafaki and Ghanbari [26] proposed a way to select θ_k such that the direction d_k^{HCG} is as close as possible to d_k^{TTHS} in the sense that their distance is minimized, i.e., the optimal choice θ_k^* is obtained by solving the least-squares problem

$$\theta_k^* = \arg \min_{\theta_k \in [0,1]} \|d_k^{HCG} - d_k^{TTHS}\|^2. \tag{5}$$

Similarly, Babaie-Kafaki and Ghanbari [27] proposed another hybrid CG method by combining HS with DY, in which the combination coefficient is also determined by the least-squares technique (5). The numerical results in [26, 27] show that this least-squares-based approach is very efficient.

Summarizing the above discussions, we have the following two observations: (1) the three-term CG methods perform well both theoretically and numerically; (2) the least-squares technique can greatly improve the efficiency of CG methods. Putting these together, the main goal of this paper is to develop new three-term CG methods that are based on the least-squares technique. More precisely, we first propose a basic three-term CG method, namely LSTT, in which the least-squares technique well combines the advantages of two existing efficient CG methods. With the Wolfe–Powell line search, LSTT is proved to be globally convergent for uniformly convex functions. In order to obtain the global convergence property for general nonlinear functions, we further present two improved variants of the LSTT CG method. All the three methods generate sufficient descent directions independent of any line search procedure. Global convergence is also analyzed for the proposed methods. Finally, some preliminary numerical results are reported to illustrate that our methods are efficient and have advantages over two famous three-term CG methods.

The paper is organized as follows. In Sect. 2, we present the basic LSTT CG method. Global convergence of LSTT is proved in Sect. 3. Two improved variants of LSTT and their convergence analysis are given in Sect. 4. Numerical results are reported in Sect. 5. Some concluding remarks are made in Sect. 6.

2 Least-squares-based three-term (LSTT) CG method

In this section, we first derive a new three-term CG formula, and then present the corresponding CG algorithm. Our formula is based on the following modified HS (MHS) formula proposed by Hager and Zhang [28, 29]:

$$\beta_k^{\text{MHS}}(\tau_k) = \beta_k^{\text{HS}} - \tau_k \frac{\|y_{k-1}\|^2 g_k^T d_{k-1}}{(d_{k-1}^T y_{k-1})^2}, \quad (6)$$

where $\tau_k (\geq 0)$ is a parameter. The corresponding direction is then given by

$$d_k^{\text{MHS}}(\tau_k) = \begin{cases} -g_k, & \text{if } k = 0, \\ -g_k + \beta_k^{\text{MHS}}(\tau_k) d_{k-1}, & \text{if } k \geq 1. \end{cases} \quad (7)$$

Different choices of τ_k will lead to different types of CG formulas. In particular, $\beta_k^{\text{MHS}}(0) = \beta_k^{\text{HS}}$, and $\beta_k^{\text{MHS}}(2)$ is just the formula proposed in [28].

In this paper, we present a more sophisticated choice of τ_k by making use of the least-squares technique. More precisely, the optimal choice τ_k^* is determined such that the direction d_k^{MHS} is as close as possible to d_k^{TTHS} , i.e., it is generated by solving the least-squares problem

$$\tau_k^* = \arg \min_{\tau_k \in [0,1]} \|d_k^{\text{MHS}}(\tau_k) - d_k^{\text{TTHS}}\|^2. \quad (8)$$

Substituting (4) and (7) in (8), we have

$$\tau_k^* = \arg \min_{\tau_k \in [0,1]} \left\| \frac{g_k^T d_{k-1}}{d_{k-1}^T y_{k-1}} y_{k-1} - \tau_k \frac{\|y_{k-1}\|^2 g_k^T d_{k-1}}{(d_{k-1}^T y_{k-1})^2} d_{k-1} \right\|^2,$$

which implies

$$\tau_k^* = \frac{(d_{k-1}^T y_{k-1})^2}{\|y_{k-1}\|^2 \|d_{k-1}\|^2}. \tag{9}$$

Thus, from (6), we obtain

$$\beta_k^{\text{MHS}}(\tau_k^*) = \beta_k^{\text{HS}} - \frac{g_k^T d_{k-1}}{\|d_{k-1}\|^2}. \tag{10}$$

So far, it seems that the two-term direction $d_k^{\text{MHS}}(\tau_k^*)$ obtained from (9) and (10) is a “good enough” direction; however, it may not always be a descent direction of the objective function. In order to overcome this difficulty, we propose a least-squares-based three-term (LSTT) direction by augmenting a term to $d_k^{\text{MHS}}(\tau_k^*)$ as follows:

$$d_k^{\text{LSTT}} = \begin{cases} -g_k, & \text{if } k = 0, \\ -g_k + \beta_k^{\text{MHS}}(\tau_k^*) d_{k-1} - \theta_k y_{k-1}, & \text{if } k \geq 1, \end{cases} \tag{11}$$

where

$$\theta_k = \frac{g_k^T d_{k-1}}{d_{k-1}^T y_{k-1}}. \tag{12}$$

The following lemma shows that the direction d_k^{LSTT} (11) is a sufficient descent direction, which is independent of the line search used.

Lemma 1 *Let the search direction $d_k := d_k^{\text{LSTT}}$ be generated by (11). Then it satisfies the following sufficient descent condition:*

$$g_k^T d_k \leq -\|g_k\|^2. \tag{13}$$

Proof For $k = 0$, we have $d_0 = -g_0$, so it follows that $g_0^T d_0 = -\|g_0\|^2$.

For $k \geq 1$, we have

$$d_k = -g_k + \beta_k^{\text{MHS}}(\tau_k^*) d_{k-1} - \theta_k y_{k-1},$$

which along with (10) and (12) shows that

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

So the proof is completed. □

Algorithm 1: Least-squares-based three-term CG algorithm (LSTT)

- Step 0:** Choose an initial point $x_0 \in \mathbb{R}^n$ and a stopping tolerance $\epsilon > 0$. Let $k := 0$.
 - Step 1:** If $\|g_k\| \leq \epsilon$, then stop.
 - Step 2:** Compute d_k by (11).
 - Step 3:** Find α_k by some line search.
 - Step 4:** Set $x_{k+1} = x_k + \alpha_k d_k$.
 - Step 5:** Let $k := k + 1$ and go to Step 1.
-

Now, we formally present the least-squares-based three-term CG algorithm (Algorithm 1) that uses d_k^{LSTT} (11) as the search direction. Note that it reduces to the classical HS method if an exact line search is executed in Step 3.

3 Convergence analysis for uniformly convex functions

In this section, we establish the global convergence of Algorithm 1 for uniformly convex functions. The stepsize α_k at Step 3 is generated by the Wolfe–Powell line search (1) and (2). For this purpose, we first make two standard assumptions on the objective function, which are assumed to be hold throughout the rest of the paper.

Assumption 1 The level set $\Omega = \{x \in \mathbb{R}^n | f(x) \leq f(x_0)\}$ is bounded.

Assumption 2 There is an open set \mathcal{O} containing Ω , in which $f(x)$ is continuous differentiable and its gradient function $g(x)$ is Lipschitz continuous, i.e., there exists a constant $L > 0$ such that

$$\|g(x) - g(y)\| \leq L\|x - y\|, \quad \forall x, y \in \mathcal{O}. \tag{14}$$

From Assumptions 1 and 2, it is not difficult to verify that there is a constant $\gamma > 0$ such that

$$\|g(x)\| \leq \gamma, \quad \forall x \in \Omega. \tag{15}$$

The following lemma is commonly used in proving the convergence of CG methods, which is called the *Zoutendijk condition* [30].

Lemma 2 Suppose that the sequence $\{x_k\}$ of iterates is generated by Algorithm 1. If the search direction d_k satisfies $g_k^T d_k < 0$ and the stepsize α_k is calculated by the Wolfe–Powell line search (1) and (2), then we have

$$\sum_{k=0}^{\infty} \frac{(g_k^T d_k)^2}{\|d_k\|^2} < +\infty. \tag{16}$$

From Lemma 1, we know that if Algorithm 1 does not stop, then

$$g_k^T d_k \leq -\|g_k\|^2 < 0.$$

Thus, under Assumptions 1 and 2, relation (16) holds immediately for Algorithm 1.

Now, we present the global convergence of Algorithm 1 (with $\epsilon = 0$) for uniformly convex functions.

Theorem 1 *Suppose that the sequence $\{x_k\}$ of iterates is generated by Algorithm 1, and that the stepsize α_k is calculated by the Wolfe–Powell line search (1) and (2). If f is uniformly convex on the level set Ω , i.e., there exists a constant $\mu > 0$ such that*

$$(g(x) - g(y))^T(x - y) \geq \mu \|x - y\|^2, \quad \forall x, y \in \Omega, \tag{17}$$

then either $\|g_k\| = 0$ for some k , or

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

Proof If $\|g_k\| = 0$ for some k , then the algorithm stops. So, in what follows, we assume that an infinite sequence $\{x_k\}$ is generated.

According to Lipschitz condition (14), the following relation holds:

$$\|y_{k-1}\| = \|g_k - g_{k-1}\| \leq L \|x_k - x_{k-1}\| = L \|s_{k-1}\|, \tag{18}$$

where $s_{k-1} := x_k - x_{k-1}$. In addition, from (17) it follows that

$$y_k^T s_k \geq \mu \|s_k\|^2. \tag{19}$$

By combining the definition of d^k (cf. (10), (11), and (12)) with relations (18) and (19), we have

$$\begin{aligned} \|d_k\| &= \left\| -g_k + \left(\frac{g_k^T y_{k-1}}{d_{k-1}^T y_{k-1}} - \frac{g_k^T d_{k-1}}{\|d_{k-1}\|^2} \right) d_{k-1} - \frac{g_k^T d_{k-1}}{d_{k-1}^T y_{k-1}} y_{k-1} \right\| \\ &\leq \|g_k\| + \frac{\|g_k\| \|y_{k-1}\|}{d_{k-1}^T y_{k-1}} \|d_{k-1}\| + \|g_k\| + \frac{\|g_k\| \|d_{k-1}\|}{d_{k-1}^T y_{k-1}} \|y_{k-1}\| \\ &= 2\|g_k\| + 2 \frac{\|g_k\| \|y_{k-1}\| \|d_{k-1}\|}{d_{k-1}^T y_{k-1}} \\ &\leq 2\|g_k\| + 2L \frac{\|g_k\| \|s_{k-1}\|^2}{\mu \|s_{k-1}\|^2} \\ &= \left(2 + \frac{2L}{\mu} \right) \|g_k\|. \end{aligned}$$

This together with Lemma 1 and (16) shows that

$$+\infty > \sum_{k=0}^{\infty} \frac{(g_k^T d_k)^2}{\|d_k\|^2} \geq \sum_{k=0}^{\infty} \frac{\|g_k\|^4}{\|d_k\|^2} \geq \frac{1}{\omega^2} \sum_{k=0}^{\infty} \|g_k\|^4, \quad \text{with } \omega = 2 + 2L/\mu,$$

which implies that $\lim_{k \rightarrow \infty} \|g_k\| = 0$. □

4 Two improved variants of the LSTT CG method

Note that the global convergence of Algorithm 1 is established only for uniformly convex functions. In this section, we present two improved variants of Algorithm 1, which both have global convergence property for general nonlinear functions.

Algorithm 2: Improved version of LSTT algorithm (LSTT+)

Step 0: Choose an initial point $x_0 \in \mathbb{R}^n$ and a tolerance $\epsilon > 0$. Let $k := 0$.

Step 1: If $\|g_k\| \leq \epsilon$, then stop.

Step 2: Compute $d_k := d_k^{\text{LSTT+}}$ by (20).

Step 3: Find α_k by some line search.

Step 4: Set $x_{k+1} = x_k + \alpha_k d_k$.

Step 5: Let $k := k + 1$ and go to Step 1.

4.1 An improved version of LSTT (LSTT+)

In fact, the main difficulty impairing convergence for general functions is that $\beta_k^{\text{MHS}}(\tau_k^*)$ (cf. (10)) may be negative. So, similar to the strategy used in [31], we present the first modification of direction d_k^{LSTT} (11) as follows:

$$d_k^{\text{LSTT+}} = \begin{cases} -g_k + \beta_k^{\text{MHS}}(\tau_k^*)d_{k-1} - \theta_k y_{k-1}, & \text{if } k > 0 \text{ and } \beta_k^{\text{MHS}}(\tau_k^*) > 0, \\ -g_k, & \text{otherwise,} \end{cases} \tag{20}$$

where $\beta_k^{\text{MHS}}(\tau_k^*)$ and θ_k are given by (10) and (12), respectively. The corresponding algorithm is given in Algorithm 2.

Obviously, the search direction d_k generated by Algorithm 2 satisfies the sufficient descent condition (13). Therefore, if the stepsize α_k is calculated by the Wolfe–Powell line search (1) and (2), then the Zoutendijk condition (16) also holds for Algorithm 2.

The following lemma shows some other important properties about the search direction d^k .

Lemma 3 *Suppose that the sequence $\{d_k\}$ of directions is generated by Algorithm 2, and that the stepsize α_k is calculated by the Wolfe–Powell line search (1) and (2). If there is a constant $c > 0$ such that $\|g_k\| \geq c$ for any k , then*

$$d_k \neq 0 \text{ for each } k, \text{ and } \sum_{k=0}^{\infty} \|u_k - u_{k-1}\|^2 < +\infty,$$

where $\|u_k\| = d_k / \|d_k\|$.

Proof Firstly, from Lemma 1 and the fact that $\|g_k\| \geq c$, we have

$$g_k^T d_k \leq -\|g_k\|^2 \leq -c^2, \quad \forall k, \tag{21}$$

which implies that $d_k \neq 0$ for each k .

Secondly, from (16) and (21), we have

$$c^4 \sum_{k=0}^{\infty} \frac{1}{\|d_k\|^2} \leq \sum_{k=0}^{\infty} \frac{\|g_k\|^4}{\|d_k\|^2} \leq \sum_{k=0}^{\infty} \frac{(g_k^T d_k)^2}{\|d_k\|^2} < +\infty. \tag{22}$$

Now we rewrite the direction d_k in (20) as

$$d_k = -g_k + \beta_k^+ d_{k-1} - \theta_k^+ y_{k-1}, \tag{23}$$

where

$$\theta_k^+ = \begin{cases} \theta_k, & \text{if } \beta_k^{\text{MHS}}(\tau_k^*) > 0, \\ 0, & \text{otherwise,} \end{cases} \quad \text{and} \quad \beta_k^+ = \max\{\beta_k^{\text{MHS}}(\tau_k^*), 0\}.$$

Denote

$$a_k = \frac{-g_k - \theta_k^+ y_{k-1}}{\|d_k\|}, \quad b_k = \beta_k^+ \frac{\|d_{k-1}\|}{\|d_k\|}. \tag{24}$$

According to (23) and (24), it follows that

$$u_k = \frac{d_k}{\|d_k\|} = \frac{-g_k - \theta_k^+ y_{k-1} + \beta_k^+ d_{k-1}}{\|d_k\|} = a_k + b_k u_{k-1}.$$

From the fact that $\|u_k\| = 1$, we obtain

$$\|a_k\| = \|u_k - b_k u_{k-1}\| = \|b_k u_k - u_{k-1}\|.$$

Since $b_k \geq 0$, we get

$$\begin{aligned} \|u_k - u_{k-1}\| &\leq \|(1 + b_k)(u_k - u_{k-1})\| \\ &\leq \|u_k - b_k u_{k-1}\| + \|b_k u_k - u_{k-1}\| \\ &= 2\|a_k\|. \end{aligned} \tag{25}$$

On the other hand, from the Wolfe–Powell line search condition (2) and (21), we have

$$d_{k-1}^T y_{k-1} = d_{k-1}^T (g_k - g_{k-1}) \geq (1 - \sigma)(-d_{k-1}^T g_{k-1}) \geq (1 - \sigma)c^2 > 0. \tag{26}$$

Since $g_{k-1}^T d_{k-1} < 0$, we have

$$g_k^T d_{k-1} = d_{k-1}^T y_{k-1} + g_{k-1}^T d_{k-1} < d_{k-1}^T y_{k-1}.$$

This together with (26) shows that

$$\frac{g_k^T d_{k-1}}{d_{k-1}^T y_{k-1}} < 1. \tag{27}$$

Again from (2), it follows that

$$g_k^T d_{k-1} \geq \sigma g_{k-1}^T d_{k-1} = -\sigma y_{k-1}^T d_{k-1} + \sigma g_k^T d_{k-1},$$

which implies

$$\frac{g_k^T d_{k-1}}{d_{k-1}^T y_{k-1}} \geq \frac{-\sigma}{1 - \sigma}. \tag{28}$$

By combining (27) and (28), we have

$$\left| \frac{g_k^T d_{k-1}}{d_{k-1}^T y_{k-1}} \right| \leq \max \left\{ \frac{\sigma}{1-\sigma}, 1 \right\}. \tag{29}$$

In addition, the following relation comes directly from (15)

$$\|y_{k-1}\| = \|g_k - g_{k-1}\| \leq \|g_k\| + \|g_{k-1}\| \leq 2\gamma. \tag{30}$$

Finally, from (15), (29), and (30), we give a bound on the numerator of a_k :

$$\begin{aligned} \|-g_k - \theta_k^+ y_{k-1}\| &\leq \|g_k\| + \left| \frac{g_k^T d_{k-1}}{d_{k-1}^T y_{k-1}} \right| \|y_{k-1}\| \\ &\leq \|g_k\| + \max \left\{ \frac{\sigma}{1-\sigma}, 1 \right\} \|y_{k-1}\| \\ &\leq M, \end{aligned}$$

where $M = \gamma + 2\gamma \max\{\frac{\sigma}{1-\sigma}, 1\}$. This together with (25) shows that

$$\|u_k - u_{k-1}\|^2 \leq 4\|a_k\|^2 \leq \frac{4M^2}{\|d_k\|^2}.$$

Summing the above relation over k and using (22), the proof is completed. □

We are now ready to prove the global convergence of Algorithm 2.

Theorem 2 *Suppose that the sequence $\{x_k\}$ of iterates is generated by Algorithm 2, and that the stepsize α_k is calculated by the Wolfe–Powell line search (1) and (2). Then either $\|g_k\| = 0$ for some k or*

$$\liminf_{k \rightarrow \infty} \|g_k\| = 0.$$

Proof Suppose by contradiction that there is a constant $c > 0$ such that $\|g_k\| \geq c$ for any k . So the conditions of Lemma 3 hold.

We first show that there is a bound on the steps s_k , whose proof is a modified version of [28, Thm. 3.2]. From Assumption 1, there is a constant $B > 0$ such that

$$\|x_k\| \leq B, \quad \forall k,$$

which implies

$$\|x_l - x_k\| \leq \|x_l\| + \|x_k\| \leq 2B. \tag{31}$$

For any $l \geq k$, it is clear that

$$x_l - x_k = \sum_{j=k}^{l-1} (x_{j+1} - x_j) = \sum_{j=k}^{l-1} \|s_j\| u_j = \sum_{j=k}^{l-1} \|s_j\| u_k + \sum_{j=k}^{l-1} \|s_j\| (u_j - u_k).$$

This together with the triangle inequality and (31) shows that

$$\sum_{j=k}^{l-1} \|s_j\| \leq \|x_l - x_k\| + \sum_{j=k}^{l-1} \|s_j\| \|u_j - u_k\| \leq 2B + \sum_{j=k}^{l-1} \|s_j\| \|u_j - u_k\|. \tag{32}$$

Denote

$$\xi := \frac{2\gamma L}{(1 - \sigma)c^2},$$

where $\sigma, L,$ and γ are given in (2), (14), and (15), respectively. Let Δ be a positive integer, chosen large enough that

$$\Delta \geq 8\xi B. \tag{33}$$

Moreover, from Lemma 3, we can choose an index k_0 large enough that

$$\sum_{i \geq k_0} \|u_{i+1} - u_i\|^2 \leq \frac{1}{4\Delta}. \tag{34}$$

Thus, if $j > k \geq k_0$ and $j - k \leq \Delta$, we can derive the following relations by (34) and the Cauchy–Schwarz inequality:

$$\begin{aligned} \|u_j - u_k\| &\leq \sum_{i=k}^{j-1} \|u_{i+1} - u_i\| \\ &\leq \sqrt{j-k} \left(\sum_{i=k}^{j-1} \|u_{i+1} - u_i\|^2 \right)^{\frac{1}{2}} \\ &\leq \sqrt{\Delta} \left(\frac{1}{4\Delta} \right)^{\frac{1}{2}} = \frac{1}{2}. \end{aligned} \tag{35}$$

Combining (32) and (35), we have

$$\sum_{j=k}^{l-1} \|s_j\| \leq 4B, \tag{36}$$

where $l > k \geq k_0$ and $l - k \leq \Delta$.

Next, we prove that there is a bound on the directions d_k .

If $d_k = -g_k$ in (20), then from (15) we have

$$\|d_k\| \leq \gamma. \tag{37}$$

In what follows, we consider the case where

$$d_k = -g_k + \beta_k^{\text{MHS}}(\tau_k^*)d_{k-1} - \theta_k y_{k-1}.$$

Thus, from (15), (18), and (26), we have

$$\begin{aligned} \|d_k\|^2 &= \left\| -g_k + \left(\frac{g_k^T y_{k-1}}{d_{k-1}^T y_{k-1}} - \frac{g_k^T d_{k-1}}{\|d_{k-1}\|^2} \right) d_{k-1} - \frac{g_k^T d_{k-1}}{d_{k-1}^T y_{k-1}} y_{k-1} \right\|^2 \\ &\leq \left(\|g_k\| + \frac{\|g_k\| \|y_{k-1}\|}{d_{k-1}^T y_{k-1}} \|d_{k-1}\| + \|g_k\| + \frac{\|g_k\| \|d_{k-1}\|}{d_{k-1}^T y_{k-1}} \|y_{k-1}\| \right)^2 \\ &= \left(2\|g_k\| + 2 \frac{\|g_k\| \|y_{k-1}\| \|d_{k-1}\|}{d_{k-1}^T y_{k-1}} \right)^2 \\ &\leq \left(2\gamma + \frac{2\gamma L}{(1-\sigma)c^2} \|s_{k-1}\| \|d_{k-1}\| \right)^2 \\ &\leq 8\gamma^2 + 2\xi^2 \|s_{k-1}\|^2 \|d_{k-1}\|^2. \end{aligned}$$

Then, by defining $S_j = 2\xi^2 \|s_j\|^2$, for $l > k_0$, we have

$$\|d_l\|^2 \leq 8\gamma^2 \left(\sum_{i=k_0+1}^l \prod_{j=i}^{l-1} S_j \right) + \|d_{k_0}\|^2 \prod_{j=k_0}^{l-1} S_j. \tag{38}$$

From (36), following the corresponding lines in [28, Thm. 3.2], we can conclude that the right-hand side of (38) is bounded, and the bound is independent of l . This together with (37) contradicts (22). Therefore, $\liminf_{k \rightarrow \infty} \|g_k\| = 0$. \square

4.2 A modified version of LSTT+ (MLSTT+)

In order to further improve the efficiency of Algorithm 2, we propose a modified version of $d_k^{\text{LSTT+}}$ (20) as follows:

$$d_k^{\text{MLSTT+}} = \begin{cases} -g_k + \beta_k^{\text{MLSTT+}} d_{k-1} - \theta_k z_{k-1}, & \text{if } k > 0 \text{ and } \beta_k^{\text{MLSTT+}} > 0, \\ -g_k, & \text{otherwise,} \end{cases} \tag{39}$$

where θ_k is given by (12) and

$$\beta_k^{\text{MLSTT+}} = \frac{g_k^T z_{k-1}}{d_{k-1}^T y_{k-1}} - \frac{g_k^T d_{k-1}}{\|d_{k-1}\|^2}, \tag{40}$$

$$z_{k-1} = g_k - \frac{\|g_k\|}{\|g_{k-1}\|} g_{k-1}. \tag{41}$$

The difference between (20) and (39) is that y_{k-1} is replaced by z_{k-1} . This idea, which aims to improve the famous PRP method, originated from [32]. Such a substitution seems useful here in that it could increase the possibility of the CG parameter being positive, and as a result, the three-term direction is used more often. In fact, as iterations go along, $\|g_k\|$ approaches zero asymptotically, and therefore the fact that $\|g_k\|/\|g_{k-1}\| < 1$ may frequently happen. If in addition $g_k^T g_{k-1} > 0$, then we have

$$g_k^T z_{k-1} = \|g_k\| - \frac{\|g_k\|}{\|g_{k-1}\|} g_k^T g_{k-1} > \|g_k\| - g_k^T g_{k-1} = g_k^T y_{k-1}.$$

The following lemma shows that the search direction (39) also has sufficient descent property.

Algorithm 3: A modified version of LSTT+ algorithm (MLSTT+)

Step 0: Choose an initial point $x_0 \in \mathbb{R}^n$ and a tolerance $\epsilon > 0$. Let $k := 0$.

Step 1: If $\|g_k\| \leq \epsilon$, then stop.

Step 2: Compute $d_k := d_k^{\text{MLSTT+}}$ by (39).

Step 3: Find α_k by the Wolfe–Powell line search (1) and (2).

Step 4: Set $x_{k+1} = x_k + \alpha_k d_k$.

Step 5: Let $k := k + 1$ and go to Step 1.

Lemma 4 *Let the search direction d_k be generated by (39). Then it satisfies the following sufficient descent condition (independent of line search):*

$$g_k^T d_k \leq -\|g_k\|^2. \tag{42}$$

Proof The proof is similar to that of Lemma 1. □

From Lemma 4, we know that the Zoutendijk condition (16) also holds for Algorithm 3. In what follows, we show that Algorithm 3 is globally convergent for general functions. The following lemma illustrates that the direction d_k generated by Algorithm 3 inherits some useful properties of $d_k^{\text{LSTT+}}$ (20), whose proof is a modification of Lemma 3.

Lemma 5 *Suppose that the sequence $\{d_k\}$ of directions is generated by Algorithm 3. If there is a constant $c > 0$ such that $\|g_k\| \geq c$ for any k , then*

$$d_k \neq 0 \text{ for each } k, \text{ and } \sum_{k=0}^{\infty} \|u_k - u_{k-1}\|^2 < +\infty,$$

where $\|u_k\| = d_k / \|d_k\|$.

Proof From the related analysis in Lemma 3, we have

$$c^4 \sum_{k=0}^{\infty} \frac{1}{\|d_k\|^2} < +\infty. \tag{43}$$

Now we redisplay the direction d_k in (39) as

$$d_k = -g_k - \hat{\theta}_k^+ z_{k-1} + \hat{\beta}_k^+ d_{k-1}, \tag{44}$$

where

$$\hat{\theta}_k^+ = \begin{cases} \theta_k, & \text{if } \beta_k^{\text{MLSTT+}} > 0, \\ 0, & \text{otherwise,} \end{cases} \text{ and } \hat{\beta}_k^+ = \max\{\beta_k^{\text{MLSTT+}}, 0\}. \tag{45}$$

Define

$$\hat{a}_k = \frac{-g_k - \hat{\theta}_k^+ z_{k-1}}{\|d_k\|}, \quad \hat{b}_k = \hat{\beta}_k^+ \frac{\|d_{k-1}\|}{\|d_k\|}. \tag{46}$$

According to (44) and (46), it follows that

$$u_k = \frac{d_k}{\|d_k\|} = \frac{-g_k - \hat{\theta}_k^+ z_{k-1} + \hat{\beta}_k^+ d_{k-1}}{\|d_k\|} = \hat{a}_k + \hat{b}_k u_{k-1}.$$

Thus, following the lines in the proof of Lemma 3, we get

$$\|u_k - u_{k-1}\| \leq 2\|\hat{a}_k\|. \tag{47}$$

Moreover, we also have

$$\left| \frac{g_k^T d_{k-1}}{d_{k-1}^T y_{k-1}} \right| \leq \max \left\{ \frac{\sigma}{1-\sigma}, 1 \right\} \quad \text{and} \quad \|y_{k-1}\| \leq 2\gamma. \tag{48}$$

The following relations hold by the definition of z_{k-1} (41):

$$\begin{aligned} \|z_{k-1}\| &\leq \|g_k - g_{k-1}\| + \left\| g_{k-1} - \frac{\|g_k\|}{\|g_{k-1}\|} g_{k-1} \right\| \\ &= \|y_{k-1}\| + \left| 1 - \frac{\|g_k\|}{\|g_{k-1}\|} \right| \|g_{k-1}\| \\ &\leq \|y_{k-1}\| + \|g_{k-1} - g_k\| \\ &= 2\|y_{k-1}\|. \end{aligned} \tag{49}$$

By combining (15), (48), and (49), we put a bound on the numerator of $\|\hat{a}_k\|$:

$$\begin{aligned} \|-g_k - \hat{\theta}_k^+ z_{k-1}\| &\leq \|g_k\| + \left| \frac{g_k^T d_{k-1}}{d_{k-1}^T y_{k-1}} \right| \|z_{k-1}\| \\ &\leq \|g_k\| + 2 \max \left\{ \frac{\sigma}{1-\sigma}, 1 \right\} \|y_{k-1}\| \\ &\leq \hat{M}, \end{aligned}$$

where $\hat{M} = \gamma + 4\gamma \max\{\frac{\sigma}{1-\sigma}, 1\}$. This together with (47) shows that

$$\|u_k - u_{k-1}\|^2 \leq 4\|\hat{a}_k\|^2 \leq \frac{4\hat{M}^2}{\|d_k\|^2}.$$

Summing the above inequalities over k and utilizing (43), we complete the proof. □

We finally present the global convergence of Algorithm 3.

Theorem 3 *Suppose that the sequence $\{x_k\}$ of iterates is generated by Algorithm 3. Then either $\|g_k\| = 0$ for some k or*

$$\liminf_{k \rightarrow \infty} \|g_k\| = 0.$$

Proof Given that there is a constant $c > 0$ such that $\|g_k\| \geq c$ for any k , then the conclusions of Lemma 5 hold.

Without loss of generality, we only consider the case where

$$d_k = -g_k + \beta_k^{\text{MLSTT}^+} d_{k-1} - \theta_k z_{k-1}.$$

So from (15), (18), (26), and (49), we obtain

$$\begin{aligned} \|d_k\|^2 &= \left\| -g_k + \left(\frac{g_k^T z_{k-1}}{d_{k-1}^T y_{k-1}} - \frac{g_k^T d_{k-1}}{\|d_{k-1}\|^2} \right) d_{k-1} - \frac{g_k^T d_{k-1}}{d_{k-1}^T y_{k-1}} z_{k-1} \right\|^2 \\ &\leq \left(\|g_k\| + \frac{\|g_k\| \|z_{k-1}\|}{d_{k-1}^T y_{k-1}} \|d_{k-1}\| + \|g_k\| + \frac{\|g_k\| \|d_{k-1}\|}{d_{k-1}^T y_{k-1}} \|z_{k-1}\| \right)^2 \\ &= \left(2\|g_k\| + 2 \frac{\|g_k\| \|z_{k-1}\| \|d_{k-1}\|}{d_{k-1}^T y_{k-1}} \right)^2 \\ &\leq \left(2\gamma + \frac{4\gamma L}{(1-\sigma)c^2} \|s_{k-1}\| \|d_{k-1}\| \right)^2 \\ &\leq 2\eta^2 + 2\rho^2 \|s_{k-1}\|^2 \|d_{k-1}\|^2, \end{aligned}$$

where $\eta = 2\gamma$ and $\rho = \frac{4\gamma L}{(1-\sigma)c^2}$.

The remainder of the argument is analogous to that of Theorem 2, hence omitted here. \square

5 Numerical results

In this section, we aim to test the practical effectiveness of Algorithm 2 (LSTT+) and Algorithm 3 (MLSTT+) which are both convergent for general functions under the Wolfe–Powell line search. The numerical results are compared with the TTPRP [22] method and the TTHS [23] method by solving 104 test problems from the CUTE library [33–35], whose dimensions range from 2 to 5,000,000.

All codes were written in Matlab R2014a and run on a PC with 4 GB RAM memory and Windows 7 operating system. The stepsizes α_k are generated by the Wolfe–Powell line search with $\sigma = 0.1$ and $\delta = 0.01$. In Tables 1, 2, 3, “Name” and “n” mean the abbreviation of the test problem and its dimension. “Itr/NF/NG” stand for the number of iterations, function evaluations, and gradient evaluations, respectively. “Tcpu” and “ $\|g_*\|$ ” denote the computing time of CPU and the final norm of the gradient value, respectively. The stopping criterion is $\|g_k\| \leq 10^{-6}$ or Itr > 2000.

To clearly show the difference in numerical effects between the above mentioned four CG methods, we present the performance profiles introduced by Dolan and Morè [36] in Figs. 1, 2, 3, 4 (with respect to Itr, NF, NG, and Tcpu, respectively), which is based on the following.

Denote the whole set of n_p test problems by \mathcal{P} , and the set of solvers by \mathcal{S} . Let $t_{p,s}$ be the Tcpu (the Itr or others) required to solve problem $p \in \mathcal{P}$ by solver $s \in \mathcal{S}$, and define the performance ratio as

$$r_{p,s} = t_{p,s} / \min_{s \in \mathcal{S}} t_{p,s}.$$

For $t_{p,s}$ of the “NaN” in Tables 1, 2, 3, we let $r_{p,s} = 2 \max\{r_{p,s} : s \in \mathcal{S}\}$, then the performance profile for each solver can be defined by

$$\rho_s(\tau) = \frac{1}{n_p} \text{size}(\{p \in \mathcal{P} : \log_2 r_{p,s} \leq \tau\}),$$

Table 1 Numerical comparisons of four CG methods

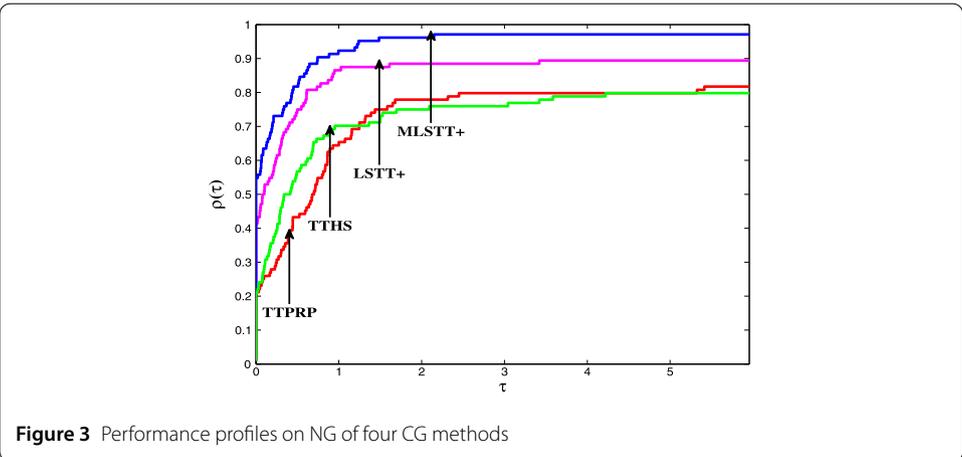
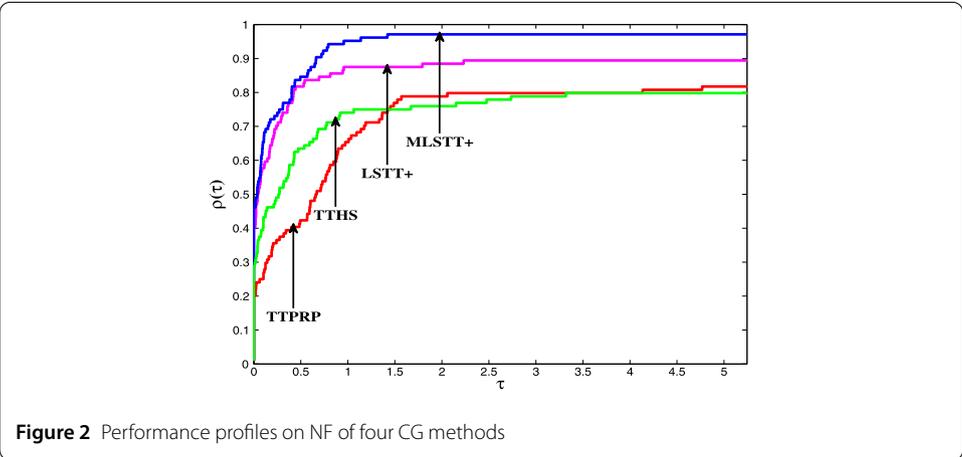
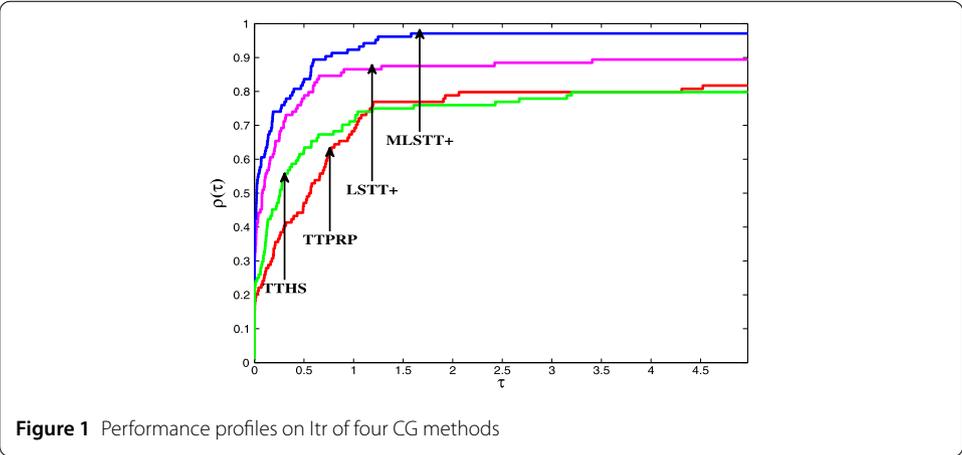
Problems	LST+	MLST+	TTPP	TTHS
Name/n	ltr/NF/NG/lcpu/lg*ll	ltr/NF/NG/lcpu/lg*ll	ltr/NF/NG/lcpu/lg*ll	ltr/NF/NG/lcpu/lg*ll
bodexp 1000	3/2/3/0.023/2.19e-108	3/2/3/0.002/2.19e-108	3/2/3/0.009/3.49e-107	3/2/3/0.002/3.45e-107
bodexp 5000	3/2/3/0.010/9.67e-110	3/2/3/0.009/9.67e-110	3/2/3/0.009/1.71e-109	3/2/3/0.009/1.71e-109
nnbdexp 10,000	3/2/3/0.017/8.32e-110	3/2/3/0.017/8.32e-110	3/2/3/0.017/1.11e-109	3/2/3/0.017/1.11e-109
exdenschmf 3000	29/81/30/0.054/3.80e-07	26/78/26/0.050/3.42e-07	31/85/34/0.057/1.15e-07	32/77/32/0.054/9.47e-07
exdenschmf 4000	24/79/25/0.063/1.94e-07	26/78/26/0.066/7.04e-07	31/85/34/0.075/1.37e-07	34/79/34/0.074/4.84e-07
exdenschmf 5000	27/81/28/0.084/2.46e-07	27/79/27/0.084/1.96e-07	31/85/34/0.092/1.51e-07	34/78/34/0.091/6.05e-07
exdenschmb 1000	26/54/27/0.008/2.81e-07	21/57/22/0.007/9.70e-08	39/58/40/0.010/4.23e-07	20/58/21/0.007/8.90e-07
exdenschmb 2000	26/54/27/0.013/3.72e-07	21/57/22/0.012/1.36e-07	39/58/40/0.018/6.02e-07	21/59/22/0.012/5.34e-07
himmelbg 10,000	3/6/7/0.016/3.75e-28	3/6/7/0.016/3.75e-28	3/6/7/0.015/4.90e-28	3/6/7/0.015/4.89e-28
himmelbg 20,000	3/6/7/0.030/5.30e-28	3/6/7/0.030/5.30e-28	3/6/7/0.030/6.93e-28	3/6/7/0.030/6.92e-28
himmelbg 25,000	3/6/7/0.038/5.92e-28	3/6/7/0.038/5.92e-28	3/6/7/0.037/7.75e-28	3/6/7/0.037/7.74e-28
genquantic 100,000	43/85/53/1.049/2.95e-07	29/64/29/0.741/5.84e-07	64/118/91/1.564/8.98e-07	55/112/82/1.440/2.75e-07
genquantic 1,000,000	45/94/57/12.877/6.87e-07	39/87/49/11.799/5.47e-07	76/162/122/22.901/6.28e-07	57/112/79/16.166/9.91e-07
genquantic 5,000,000	44/96/57/65.600/3.38e-07	45/84/51/62.282/3.12e-08	81/185/138/126.781/1.50e-07	42/109/61/67.433/4.05e-07
biggsb1 200	802/376/967/0.077/9.19e-07	671/349/823/0.065/6.83e-07	1105/827/1496/0.108/9.02e-07	729/328/871/0.065/8.56e-07
biggsb1 400	1726/877/2143/0.203/9.01e-07	1632/841/2031/0.201/8.39e-07	1992/1465/2703/0.246/8.02e-07	Na/Na/Na/Na/Na/4.04e-06
sine 30,000	39/59/41/0.556/7.11e-07	130/158/182/1.663/1.37e-07	971/1607/1748/12.793/4.81e-07	304/328/439/3.544/4.74e-07
sine 50,000	41/109/70/1.359/8.18e-07	131/187/196/2.916/5.26e-07	Na/Na/Na/Na/Na/2.87e-05	363/484/577/7.323/9.63e-07
sine 200,000	189/316/320/17.825/1.92e-07	114/164/168/11.351/4.66e-07	Na/Na/Na/Na/Na/5.36e-05	1.227/1.1636/2017/104.296/6.06e-07
sinquad 3	38/67/51/0.004/9.88e-07	60/116/94/0.006/1.91e-07	77/193/153/0.008/3.34e-07	65/81/82/0.005/8.11e-07
fletcbv3 50	97/113/152/0.011/2.19e-07	139/172/219/0.016/6.63e-07	159/185/250/0.019/2.37e-07	134/164/211/0.016/1.49e-07
fletcbv3 100	490/414/684/0.099/3.17e-07	899/612/1181/0.175/3.45e-07	348/367/517/0.074/4.28e-07	Na/Na/Na/Na/Na/7.69e-04
eq2 20	32/88/49/0.004/9.14e-07	51/119/82/0.006/5.52e-07	62/247/157/0.009/9.50e-07	96/280/209/0.012/7.86e-07
nonscomp 20,000	79/110/96/0.2398/6.66e-07	61/89/68/0.239/7.88e-07	140/153/178/0.492/8.47e-07	142/142/175/0.484/6.22e-07
nonscomp 30,000	55/93/64/0.325/7.66e-07	66/83/70/0.361/9.01e-07	81/123/105/0.470/7.86e-07	149/141/181/0.742/8.09e-07
nonscomp 50,000	66/86/70/0.585/5.09e-07	61/91/69/0.601/7.01e-07	89/121/110/0.822/7.51e-07	77/107/92/0.708/4.46e-07
cosine 5000	39/72/49/0.093/9.27e-07	23/59/26/0.060/3.61e-07	90/158/142/0.227/9.66e-07	57/89/75/0.121/3.61e-07
cosine 10,000	28/60/31/0.131/2.38e-07	21/62/25/0.115/7.31e-07	517/1054/1002/2.277/6.53e-07	61/93/81/0.236/9.26e-07
cosine 1,000,000	145/277/257/72.095/8.55e-09	22/59/24/13.451/8.68e-07	Na/Na/Na/Na/Na/3.07e-04	277/393/447/119.491/5.17e-07
dixmaana 3000	25/64/26/0.207/2.49e-07	16/64/17/0.174/3.25e-07	20/63/21/0.186/1.56e-07	18/65/19/0.183/1.49e-07
dixmaand 3000	17/63/17/0.177/5.34e-07	13/57/13/0.149/8.63e-07	17/65/17/0.179/9.00e-07	20/66/20/0.191/2.94e-07
dixmaane 3000	357/227/440/1.805/7.00e-07	374/252/469/1.939/4.33e-07	264/231/349/1.472/7.23e-07	436/305/557/2.279/9.23e-07
dixmaang 3000	296/210/369/1.541/6.80e-07	344/217/420/1.741/5.51e-07	326/249/418/1.741/8.47e-07	327/164/377/1.543/9.26e-07
dixmaanl 3000	972/551/1217/4.699/9.58e-07	1131/542/1372/5.284/9.09e-07	960/715/1286/5.035/8.43e-07	1250/597/1518/5.800/9.04e-07
dixmaanl 3000	983/544/1222/4.736/8.17e-07	Na/Na/Na/Na/Na/8.19e-06	Na/Na/Na/Na/Na/1.73e-05	Na/Na/Na/Na/Na/1.46e-05

Table 2 Numerical comparisons of four CG methods (continued)

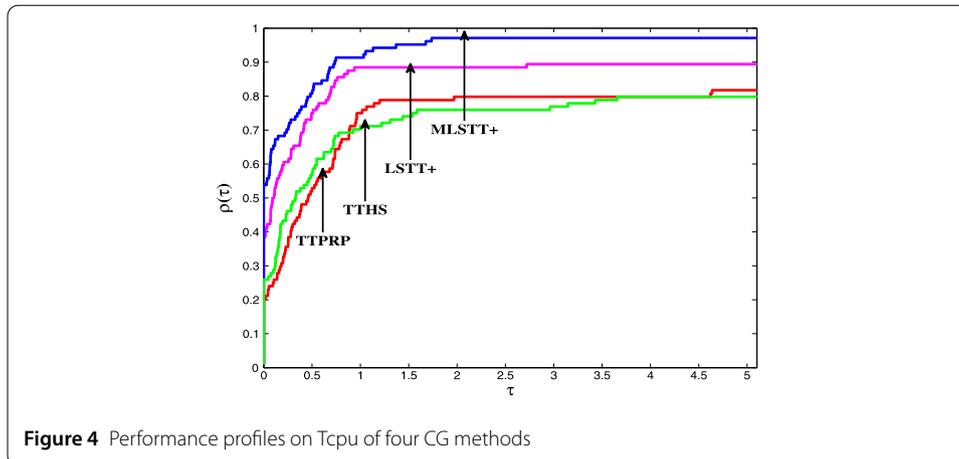
Problems	LST+	MLST+	TTPR	TTSS
Name/n	$\text{itr}/\text{NF}/\text{NG}/\text{fcpu}/\ \text{g}\ $	$\text{itr}/\text{NF}/\text{NG}/\text{fcpu}/\ \text{g}\ $	$\text{itr}/\text{NF}/\text{NG}/\text{fcpu}/\ \text{g}\ $	$\text{itr}/\text{NF}/\text{NG}/\text{fcpu}/\ \text{g}\ $
dixon3dq 10	83/91/105/0.007/9.25e-07	87/71/101/0.007/6.21e-07	99/107/129/0.008/9.56e-07	95/76/109/0.007/6.39e-07
dixon3dq 100	537/290/659/0.044/8.49e-07	894/459/1100/0.075/9.34e-07	604/430/796/0.051/8.54e-07	673/316/808/0.052/8.86e-07
ddgrtc 300	73/162/115/0.011/2.42e-07	45/121/67/0.007/5.82e-07	94/206/159/0.014/4.48e-07	74/162/118/0.011/9.63e-07
ddgrtc 600	73/124/97/0.014/9.95e-07	67/164/111/0.015/8.03e-07	77/188/132/0.017/7.82e-07	75/145/109/0.015/8.36e-07
ddgrtc 100	28/74/28/0.007/5.04e-07	24/72/24/0.006/6.13e-07	23/73/23/0.006/5.74e-07	27/74/28/0.006/4.51e-07
ddgrtc 500	40/102/47/0.030/9.16e-07	37/98/42/0.029/3.25e-07	37/91/38/0.027/5.75e-07	41/93/44/0.029/5.29e-07
edensch 1000	47/129/83/0.084/5.30e-07	33/168/88/0.091/3.13e-07	51/291/148/0.153/9.32e-07	NaN/NaN/NaN/NaN/5.95e-06
edensch 2500	NaN/NaN/NaN/NaN/1.21e-05	32/134/57/0.173/4.30e-07	56/558/284/0.650/8.30e-07	NaN/NaN/NaN/NaN/9.76e-06
edensch 3500	NaN/NaN/NaN/NaN/2.89e-06	30/160/74/0.278/6.62e-07	NaN/NaN/NaN/NaN/8.18e-06	NaN/NaN/NaN/NaN/5.88e-06
engvall 10	NaN/NaN/NaN/NaN/1.16e+00	23/61/23/0.003/6.70e-08	NaN/NaN/NaN/NaN/1.94e+00	NaN/NaN/NaN/NaN/7.79e-01
erimos 10	NaN/NaN/NaN/NaN/3.26e-06	671/840/1051/0.075/9.13e-07	NaN/NaN/NaN/NaN/2.20e-02	NaN/NaN/NaN/NaN/2.13e-06
fletcher 50	33/73/43/0.004/9.44e-07	42/85/58/0.005/9.43e-07	NaN/NaN/NaN/NaN/1.88e-06	NaN/NaN/NaN/NaN/3.57e-06
fletcher 100	50/85/65/0.006/9.18e-07	44/72/55/0.005/8.43e-07	51/83/65/0.006/9.63e-07	49/135/89/0.007/4.32e-07
fletcher 10,000	NaN/NaN/NaN/NaN/4.92e-04	62/113/91/0.176/9.17e-07	NaN/NaN/NaN/NaN/6.18e-04	NaN/NaN/NaN/NaN/5.24e-05
freuroth 100	NaN/NaN/NaN/NaN/2.12e-04	NaN/NaN/NaN/NaN/1.80e-05	NaN/NaN/NaN/NaN/7.32e-05	NaN/NaN/NaN/NaN/3.12e-06
genrose 6000	110/129/142/0.177/7.99e-07	108/139/144/0.186/8.45e-07	180/202/247/0.289/7.85e-07	175/167/225/0.264/4.41e-07
genrose 10,000	139/186/199/0.388/9.81e-07	220/222/298/0.577/2.74e-07	200/210/271/0.511/9.58e-07	190/184/249/0.471/8.55e-07
genrose 15,000	118/132/150/0.447/4.01e-07	165/196/229/0.672/7.76e-07	197/210/268/0.750/4.91e-07	195/178/249/0.700/4.54e-07
liarwhd 1000	84/242/163/0.045/9.07e-07	115/320/232/0.063/4.57e-07	NaN/NaN/NaN/NaN/8.71e-06	NaN/NaN/NaN/NaN/5.84e-05
liarwhd 2000	116/304/224/0.097/4.36e-07	109/299/217/0.097/2.56e-07	NaN/NaN/NaN/NaN/3.96e-01	NaN/NaN/NaN/NaN/3.34e-03
liarwhd 30,000	NaN/NaN/NaN/NaN/9.33e+01	332/966/766/3.672/8.26e-07	NaN/NaN/NaN/NaN/3.51e+02	NaN/NaN/NaN/NaN/4.97e+02
nondquar 100	NaN/NaN/NaN/NaN/7.30e-05	NaN/NaN/NaN/NaN/2.59e-04	NaN/NaN/NaN/NaN/2.52e-04	NaN/NaN/NaN/NaN/3.45e-04
penalty 1 500	13/72/13/0.095/6.77e-07	13/72/13/0.094/6.77e-07	13/72/13/0.094/6.77e-07	13/72/13/0.093/6.77e-07
penalty 1 8000	12/81/13/4.565/9.61e-07	12/81/13/4.556/9.61e-07	12/81/13/4.553/9.61e-07	12/81/13/4.554/9.61e-07
penalty 1 5000	17/79/17/11.321/3.64e-07	17/79/17/11.281/3.57e-07	17/79/17/11.251/3.51e-07	17/79/17/11.271/4.16e-07
power1 30	307/222/381/0.023/7.59e-07	292/213/362/0.022/9.61e-07	303/218/374/0.022/7.94e-07	318/172/367/0.022/7.35e-07
quartic 100	28/74/28/0.007/5.04e-07	24/72/24/0.006/6.13e-07	23/73/23/0.006/5.74e-07	27/74/28/0.006/4.51e-07
quartic 400	33/92/36/0.021/7.12e-07	37/95/40/0.023/9.77e-08	31/89/31/0.020/2.76e-07	34/94/37/0.022/1.27e-07
tridia 100	273/200/338/0.026/8.84e-07	248/185/306/0.024/9.47e-07	429/348/568/0.041/8.73e-07	302/173/353/0.027/8.98e-07
tridia 1500	1171/632/1444/0.404/6.91e-07	1591/856/1976/0.572/8.12e-07	1615/1212/2178/0.600/7.67e-07	1424/657/1710/0.471/9.70e-07
ntridia 3000	1809/987/2258/1.136/8.46e-07	1900/966/2338/1.231/6.52e-07	NaN/NaN/NaN/NaN/9.82e-06	1959/929/2379/1.186/6.39e-07
raydan1 1000	260/181/319/0.067/9.99e-07	288/286/400/0.076/9.72e-07	NaN/NaN/NaN/NaN/1.68e-05	347/230/431/0.079/5.61e-07
raydan1 4000	NaN/NaN/NaN/NaN/1.61e-06	931/1190/1494/0.699/9.15e-07	NaN/NaN/NaN/NaN/1.44e-04	NaN/NaN/NaN/NaN/8.98e-05
raydan2 3000	17/52/17/0.022/6.83e-07	17/52/17/0.022/6.83e-07	17/52/17/0.022/6.82e-07	17/52/17/0.022/6.82e-07
raydan2 10,000	17/56/20/0.071/3.89e-07	20/63/25/0.084/9.15e-07	17/80/38/0.096/7.63e-07	14/53/16/0.065/9.17e-07

Table 3 Numerical comparisons of four CG methods (continued)

Problems Name/n	LST+		MLST+		TTPRP		TTHS	
	itr/NF/NG/Tcpu/ g*	g*	itr/NF/NG/Tcpu/ g*	g*	itr/NF/NG/Tcpu/ g*	g*	itr/NF/NG/Tcpu/ g*	g*
raydan2 20,000	14/78/29/0.169/3.19e-07		17/54/19/0.139/8.35e-07		17/54/19/0.136/6.26e-07		NaN/NaN/NaN/NaN/3.32e-05	
diagonal1 60	NaN/NaN/NaN/NaN/5.01e-06		96/343/215/0.01015/9.09e-07		71/219/161/0.01010/9.47e-07		NaN/NaN/NaN/NaN/5.78e-06	
diagonal2 15,000	828/548/1076/4.042/6.21e-07		807/447/1004/3.878/7.15e-07		832/742/1176/4.435/8.85e-07		899/527/1135/4.248/5.79e-07	
diagonal2 20,000	947/676/1259/6.252/8.52e-07		1028/633/1319/6.707/8.60e-07		1075/893/1496/7.451/9.85e-07		1031/566/1288/6.366/7.35e-07	
diagonal2 50,000	1425/829/1813/21.910/8.79e-07		1504/895/1925/23.985/9.63e-07		NaN/NaN/NaN/NaN/1.68e-06		NaN/NaN/NaN/NaN/1.97e-06	
diagonal3 30	62/61/66/0.006/9.25e-07		59/64/64/0.005/5.95e-07		63/62/67/0.005/8.85e-07		61/98/78/0.006/7.47e-07	
diagonal3 100	NaN/NaN/NaN/NaN/1.05e-05		122/754/474/0.035/7.10e-07		NaN/NaN/NaN/NaN/1.10e-05		NaN/NaN/NaN/NaN/1.26e-06	
bv 300	1679/801/2072/1.434/8.06e-07		1193/589/1481/1.027/9.41e-07		1687/1256/2307/1.620/9.17e-07		NaN/NaN/NaN/NaN/2.92e-06	
bv 1500	13/10/15/0.280/7.37e-07		27/22/35/0.653/7.83e-07		24/18/30/0.554/8.09e-07		26/12/29/0.517/9.28e-07	
bv 2000	5/5/5/0.174/8.10e-07		5/5/5/0.173/8.31e-07		7/7/8/0.274/7.81e-07		7/5/7/0.232/8.75e-07	
ie 10	15/37/15/0.005/4.09e-07		9/38/9/0.004/4.36e-07		15/40/15/0.005/5.67e-07		13/39/13/0.005/4.44e-07	
ie 100	19/43/19/0.244/7.49e-07		13/42/13/0.203/2.63e-07		16/41/16/0.219/6.58e-07		19/44/19/0.248/2.70e-07	
ie 200	16/42/16/0.861/7.41e-07		14/43/14/0.818/1.22e-08		19/41/19/0.916/8.60e-07		15/44/15/0.853/5.61e-07	
singx 100	149/255/242/0.037/8.69e-07		305/420/482/0.071/9.92e-07		245/656/539/0.085/7.28e-07		248/464/447/0.064/8.44e-07	
singx 800	182/306/299/3.003/8.08e-07		137/189/196/1.992/6.79e-07		188/517/413/4.136/7.94e-07		211/395/372/3.683/7.38e-07	
singx 3000	250/433/432/60.254/7.77e-07		247/385/402/56.118/3.19e-07		262/707/578/82.289/6.97e-07		238/465/433/61.068/7.24e-07	
woods 100	308/549/539/2.939/5.14e-07		248/411/409/2.277/9.23e-07		308/721/625/3.447/4.62e-07		192/414/356/2.014/9.92e-07	
band 3	19/63/20/0.003/9.34e-07		20/66/22/0.003/2.01e-07		19/62/19/0.003/2.06e-07		19/63/19/0.003/3.26e-07	
band 50	28/68/29/0.014/5.93e-07		26/70/27/0.014/7.08e-07		48/71/49/0.019/7.50e-07		37/70/38/0.016/8.95e-07	
band 3	146/231/236/0.044/9.52e-07		84/119/116/0.024/7.55e-07		141/353/290/0.053/9.71e-07		82/165/136/0.027/1.92e-07	
beale 2	46/136/87/0.007/6.62e-07		55/123/91/0.007/5.02e-07		55/217/133/0.009/8.50e-07		42/117/73/0.005/5.77e-07	
box 3	76/246/171/0.016/8.42e-07		48/106/76/0.008/2.48e-07		56/183/124/0.011/7.91e-07		47/71/56/0.006/5.32e-07	
froth 2	92/283/204/0.014/4.19e-07		77/322/196/0.013/7.76e-07		56/400/216/0.014/9.58e-07		48/243/137/0.009/9.42e-07	
jensam 2	45/141/89/0.007/9.89e-07		66/130/103/0.008/8.59e-07		43/149/89/0.007/2.82e-07		40/169/95/0.007/3.18e-07	
kwosb 4	246/384/420/0.037/4.44e-07		189/264/304/0.028/9.95e-07		211/541/464/0.040/9.10e-07		199/452/408/0.035/9.51e-07	
lin 500	2/2/2/0.028/1.10e-13		2/2/2/0.029/1.10e-13		2/2/2/0.028/1.10e-13		2/2/2/0.028/1.10e-13	
osb2 11	547/436/741/0.138/9.70e-07		51/3/403/690/0.128/9.17e-07		620/638/914/0.165/9.72e-07		409/325/546/0.100/6.25e-07	
pen1 80	79/255/156/0.160/4.38e-07		173/495/368/0.332/1.41e-07		80/285/175/0.173/3.16e-07		89/318/197/0.194/1.38e-07	
rosex 1100	89/224/165/2.938/4.86e-07		94/253/185/3.234/9.68e-07		71/271/169/3.085/2.45e-07		56/302/174/3.146/5.25e-07	
trid 100	81/78/91/0.031/9.37e-07		77/84/90/0.031/9.50e-07		100/96/118/0.038/7.88e-07		91/77/100/0.033/9.27e-07	
trid 1000	94/81/103/1.019/7.08e-07		93/79/103/1.018/9.02e-07		96/91/110/1.077/9.02e-07		105/77/113/1.084/9.01e-07	
vardim 8	15/86/15/0.003/7.29e-08		15/86/15/0.003/7.29e-08		15/86/15/0.003/7.29e-08		15/86/15/0.003/7.29e-08	
watson 3	47/76/56/0.010/2.35e-07		49/80/61/0.010/7.77e-07		55/129/91/0.014/3.63e-07		53/121/85/0.013/4.29e-07	
wood 4	241/434/418/0.030/6.55e-07		200/348/335/0.025/6.91e-07		243/563/484/0.033/9.17e-07		132/247/217/0.016/7.56e-07	



where $\text{size}(A)$ stands for the number of elements in the set A . Hence $\rho_s(\tau)$ is the probability for solver $s \in \mathcal{S}$ that the performance ratio $r_{p,s}$ is within a factor $\tau \in \mathbb{R}$. The function ρ_s is the (cumulative) distribution function for the performance ratio. Apparently the solver whose curved shape is on the top will win over the rest of the solvers. Refer to [36] for more details.



For each method, the performance profile plots the fraction $\rho_s(\tau)$ of the problems for which the method is within a factor τ of the best time. The left side of the figure represents the percentage of the test problems for which a method is the fastest. The right side represents the percentage of the test problems that are successfully solved by each of the methods. The top curve is the method that solved the most problems in a time that was within a factor τ of the best time.

In Figs. 1, 2, 3, 4, we compare the performance of the LSTT+ method and the MLSTT+ method with the TTPRP method and the TTHS method. We observe from Fig. 1 that MLSTT+ is the fastest for about 51% of the test problems with the smallest number of iterations, and it ultimately solves about 98% of the test problems. LSTT+ has the second best performance which can solve 88% of the test problems successfully, while TTPRP and TTHS solve about 80% and 78% of the test problems successfully, respectively. Figure 2 shows that MLSTT+ exhibits the best performance for the number of function evaluations since it can solve about 49% of the test problems with the smallest number of function evaluations; LSTT+ has the second best performance as it solves about 40% in the same situation. From Fig. 3, it is not difficult to see that MLSTT+ and LSTT+ perform better than the other two methods about the number of gradient evaluations. Moreover, MLSTT+ is the fastest for the number of gradient evaluations since it solves about 56% of the test problems with the smallest number of gradient evaluations, while LSTT+ solves about 41% of the test problems with the smallest number of gradient evaluations. In Fig. 4, MLSTT+ displays the best performance for CPU time since it solves about 53% of the test problems with the least CPU time and the data for LSTT+ is 42% in the same case, which is second. Since all methods were implemented with the same line search, we can conclude that the LSTT+ method and the MLSTT+ method seem more efficient.

Combining Tables 1, 2, 3 and Figs. 1, 2, 3, 4, we are led to the conclusion that LSTT+ and MLSTT+ perform better than TTPRP and TTHS, in which MLSTT+ is the best one. This shows that the proposed methods of this paper possess good numerical performance.

6 Conclusion

In this paper, we have presented three new three-term CG methods that are based on the least-squares technique to determine the CG parameters. All can generate sufficient descent directions without the help of a line search procedure. The basic one is globally convergent for uniformly convex functions, while the other two improved variants possess

global convergence for general nonlinear functions. Preliminary numerical results show that our methods are very promising.

Acknowledgements

The authors wish to thank the two anonymous referees and the editor for their constructive and pertinent suggestions for improving both the presentation and the numerical experiments. They would like to thank for the support of funds as well.

Funding

This work was supported by the National Natural Science Foundation (11761013) and Guangxi Natural Science Foundation (2018GXNSFFA281007) of China.

Availability of data and materials

Not applicable.

Competing interests

The authors declare that they have no competing interests.

Authors' contributions

All authors read and approved the final manuscript. CT mainly contributed to the algorithm design and convergence analysis; SL mainly contributed to the convergence analysis and numerical results; and ZC mainly contributed to the algorithm design.

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Received: 10 October 2019 Accepted: 27 January 2020 Published online: 03 February 2020

References

1. Tripathi, A., McNulty, I., Shpyrko, O.G.: Ptychographic overlap constraint errors and the limits of their numerical recovery using conjugate gradient descent methods. *Opt. Express* **22**(2), 1452–1466 (2014)
2. Antoine, X., Levitt, A., Tang, Q.: Efficient spectral computation of the stationary states of rotating Bose–Einstein condensates by preconditioned nonlinear conjugate gradient methods. *J. Comput. Phys.* **343**, 92–109 (2017)
3. Azimi, A., Daneshgar, E.: Indoor contaminant source identification by inverse zonal method: Levenberg–Marquardt and conjugate gradient methods. *Adv. Build. Energy Res.* **12**(2), 250–273 (2018)
4. Yang, L.F., Jian, J.B., Wang, Y.Y., Dong, Z.Y.: Projected mixed integer programming formulations for unit commitment problem. *Int. J. Electr. Power Energy Syst.* **68**, 195–202 (2015)
5. Yang, L.F., Jian, J.B., Zhu, Y.N., Dong, Z.Y.: Tight relaxation method for unit commitment problem using reformulation and lift-and-project. *IEEE Trans. Power Syst.* **30**(1), 13–23 (2015)
6. Yang, L.F., Zhang, C., Jian, J.B., Meng, K., Xu, Y., Dong, Z.Y.: A novel projected two-binary-variable formulation for unit commitment in power systems. *Appl. Energy* **187**, 732–745 (2017)
7. Hestenes, M.R., Stiefel, E.: Methods of conjugate gradients for solving linear systems. *J. Res. Natl. Bur. Stand.* **49**(6), 409–436 (1952)
8. Fletcher, R., Reeves, C.M.: Function minimization by conjugate gradients. *Comput. J.* **7**(2), 149–154 (1964)
9. Polak, E.: Note sur la convergence de méthodes de directions conjuguées. *Revue Française Information Recherche Operationnelle* **16**(16), 35–43 (1969)
10. Polyak, B.T.: The conjugate gradient method in extremal problems. *USSR Comput. Math. Math. Phys.* **9**(4), 94–112 (1969)
11. Dai, Y.H., Yuan, Y.X.: A nonlinear conjugate gradient method with a strong global convergence property. *SIAM J. Optim.* **10**(1), 177–182 (1999)
12. Dong, X.L., Liu, H.W., He, Y.B.: New version of the three-term conjugate gradient method based on spectral scaling conjugacy condition that generates descent search direction. *Appl. Math. Comput.* **269**, 606–617 (2015)
13. Jian, J.B., Chen, Q., Jiang, X.Z., Zeng, Y.F., Yin, J.H.: A new spectral conjugate gradient method for large-scale unconstrained optimization. *Optim. Methods Softw.* **32**(3), 503–515 (2017)
14. Sun, M., Liu, J.: New hybrid conjugate gradient projection method for the convex constrained equations. *Calcolo* **53**(3), 399–411 (2016)
15. Mtagulwa, P., Kaelo, P.: An efficient modified PRP-FR hybrid conjugate gradient method for solving unconstrained optimization problems. *Appl. Numer. Math.* **145**, 111–120 (2019)
16. Dong, X.-L., Han, D.-R., Ghanbari, R., Li, X.-L., Dai, Z.-F.: Some new three-term Hestenes–Stiefel conjugate gradient methods with affine combination. *Optimization* **66**(5), 759–776 (2017)
17. Albaali, M., Narushima, Y., Yabe, H.: A family of three-term conjugate gradient methods with sufficient descent property for unconstrained optimization. *Comput. Optim. Appl.* **60**(1), 89–110 (2015)
18. Babaie-Kafaki, S., Ghanbari, R.: Two modified three-term conjugate gradient methods with sufficient descent property. *Optim. Lett.* **8**(8), 2285–2297 (2014)
19. Arzuka, I., Bakar, M.R.A., Leong, W.J.: A scaled three-term conjugate gradient method for unconstrained optimization. *J. Inequal. Appl.* **2016**(1), Article ID 325 (2016)
20. Liu, J.K., Feng, Y.M., Zou, L.M.: Some three-term conjugate gradient methods with the inexact line search condition. *Calcolo* **55**(2), Article ID 16 (2018)

21. Li, M.: A family of three-term nonlinear conjugate gradient methods close to the memoryless BFGS method. *Optim. Lett.* **12**(8), 1911–1927 (2018)
22. Zhang, L., Zhou, W.J., Li, D.H.: A descent modified Polak–Ribière–Polyak conjugate gradient method and its global convergence. *IMA J. Numer. Anal.* **26**(4), 629–640 (2006)
23. Zhang, L., Zhou, W.J., Li, D.H.: Some descent three-term conjugate gradient methods and their global convergence. *Optim. Methods Softw.* **22**(4), 697–711 (2007)
24. Dennis, J.E. Jr., Moré, J.J.: Quasi-Newton methods, motivation and theory. *SIAM Rev.* **19**(1), 46–89 (1977)
25. Zhang, L., Zhou, W.J., Li, D.H.: Global convergence of a modified Fletcher–Reeves conjugate gradient method with Armijo-type line search. *Numer. Math.* **104**(4), 561–572 (2006)
26. Babaie-Kafaki, S., Ghanbari, R.: A hybridization of the Polak–Ribière–Polyak and Fletcher–Reeves conjugate gradient methods. *Numer. Algorithms* **68**(3), 481–495 (2015)
27. Babaie-Kafaki, S., Ghanbari, R.: A hybridization of the Hestenes–Stiefel and Dai–Yuan conjugate gradient methods based on a least-squares approach. *Optim. Methods Softw.* **30**(4), 673–681 (2015)
28. Hager, W.W., Zhang, H.C.: A new conjugate gradient method with guaranteed descent and an efficient line search. *SIAM J. Optim.* **16**(1), 170–192 (2005)
29. Hager, W.W., Zhang, H.C.: A survey of nonlinear conjugate gradient methods. *Pac. J. Optim.* **2**(1), 35–58 (2006)
30. Zoutendijk, G.: Nonlinear programming, computational methods. In: Abadie, J. (ed.) *Integer and Nonlinear Programming*, pp. 37–86. North-Holland, Amsterdam (1970)
31. Gilbert, J.C., Nocedal, J.: Global convergence properties of conjugate gradient methods for optimization. *SIAM J. Optim.* **2**(1), 21–42 (1992)
32. Wei, Z.X., Yao, S.W., Liu, L.Y.: The convergence properties of some new conjugate gradient methods. *Appl. Math. Comput.* **183**(2), 1341–1350 (2006)
33. More, J.J., Garbow, B.S., Hillstom, K.E.: Testing unconstrained optimization software. *ACM Trans. Math. Softw.* **7**(1), 17–41 (1981)
34. Bongartz, I., Conn, A.R., Gould, N., Toint, P.L.: CUTE: constrained and unconstrained testing environment. *ACM Trans. Math. Softw.* **21**(1), 123–160 (1995)
35. Andrei, N.: An unconstrained optimization test functions collection. *Adv. Model. Optim.* **10**(1), 147–161 (2008)
36. Dolan, E.D., Moré, J.J.: Benchmarking optimization software with performance profiles. *Math. Program.* **91**(2), 201–213 (2002)

Submit your manuscript to a SpringerOpen[®] journal and benefit from:

- Convenient online submission
- Rigorous peer review
- Open access: articles freely available online
- High visibility within the field
- Retaining the copyright to your article

Submit your next manuscript at ► [springeropen.com](https://www.springeropen.com)
