Calcolo (2018) 55:16
https://doi.org/10.1007/s10092-018-0258-3

Some three-term conjugate gradient methods with the inexact line search condition

J. K. Liu[1] · Y. M. Feng[1] · L. M. Zou[1]

Received: 24 May 2017 / Accepted: 12 March 2018 / Published online: 16 March 2018
© Springer-Verlag Italia S.r.l., part of Springer Nature 2018

Abstract The three-term conjugate gradient methods solving large-scale optimization problems are favored by many researchers because of their nice descent and convergent properties. In this paper, we extend some new conjugate gradient methods, and construct some three-term conjugate gradient methods. An remarkable property of the proposed methods is that the search direction always satisfies the sufficient descent condition without any line search. Under the standard Wolfe line search, the global convergence properties of the proposed methods are proved merely by assuming that the objective function is Lipschitz continuous. Preliminary numerical results and comparisons show that the proposed methods are efficient and promising.

Keywords Unconstrained optimization problem · Three-term conjugate gradient method · Sufficient descent property · Global convergence

Mathematics Subject Classification 90C30 · 65K05

This research is funded by Chongqing Research Program of Basic Research and Frontier Technology (Grant No.: cstc2017jcyjAX0318), the fund of Scientific and Technological Research Program of Chongqing Municipal Education Commission (Grant Nos.: KJ1710251, KJ1501003), Program for Innovation Team Building at Institutions of Higher Education in Chongqing (Grant number: CXTDX201601035) and Project Supported by Chongqing Municipal Key Laboratory of Institutions of Higher Education (Grant No. [2017]3).

✉ J. K. Liu
liujinkui2006@126.com

[1] School of Mathematics and Statistics, Chongqing Three Gorges University, Chongqing 404100, China

1 Introduction

In this paper, we consider the large-scale unconstrained optimization problem

$$\min f(x), x \in R^n$$

where $f : R^n \to R$ is continuously differentiable. The gradient-type iterative methods are usually used to solve this problem by generating an iterative sequence $\{x_k\}$, using the following formula

$$x_{k+1} = x_k + \alpha_k d_k, \tag{1.1}$$

for $k \geq 0$, where x_0 is the initial point, $\alpha_k > 0$ is the step-length and d_k is the search direction.

The conjugate gradient method is one of the most effective gradient-type iterative methods for solving large-scale unconstrained optimization problems, and its direction is defined as

$$d_k = \begin{cases} -g_k, & if\ k = 0, \\ -g_k + \beta_k d_{k-1}, & if\ k \geq 1. \end{cases} \tag{1.2}$$

where g_k denotes the gradient $g(x_k)$ of f at point x_k, and β_k is the conjugate gradient update parameter. Different conjugate gradient methods correspond to different choices for the update parameter β_k. Well-known update parameters β_k have the Hestenes–Stiefel (HS) [1], the Fletcher–Reeves (FR) [2], the Polak–Ribière–Polyak (PRP) [3,4], the Conjugate Descent (CD) [5], the Liu-Storey [6], and the Dai–Yuan (DY) [7], given respectively by

$$\beta_k^{HS} = \frac{g_k^T y_{k-1}}{d_{k-1}^T y_{k-1}}, \qquad \beta_k^{FR} = \frac{||g_k||^2}{||g_{k-1}||^2}, \qquad \beta_k^{PRP} = \frac{g_k^T y_{k-1}}{||g_{k-1}||^2},$$
$$\beta_k^{CD} = \frac{||g_k||^2}{-d_{k-1}^T g_{k-1}}, \qquad \beta_k^{LS} = \frac{g_k^T y_{k-1}}{-d_{k-1}^T g_{k-1}}, \qquad \beta_k^{DY} = \frac{||g_k||^2}{d_{k-1}^T y_{k-1}},$$

where $y_{k-1} = g_k - g_{k-1}$ and $||\cdot||$ denotes the Euclidean norm. If f is a strongly convex quadratic function,then in theory, the above update parameters are equivalent under the exact line search. For non-quadratic functions, each choice for update parameter leads to different performance. When the algorithm moves an extremely small step-length along the search direction d_{k-1}, i.e., $x_k \approx x_{k-1}$, this implies $y_{k-1} \approx 0$. In this case, PRP, HS and LS methods essentially perform a restart. This property makes these methods generate effective numerical results for large-scale optimization problems. However, their global convergence is not still proven under some inexact line searches (e.g., Wolfe-type line search). One of the main reason is that the search direction d_k is not descent for general objective functions with some inexact line searches. But, in practice, it generally requires too many evaluations of the objective function f and possibly the gradient ∇f to generate a step-length in per-iteration by using the exact line search. More practical strategies perform an inexact line search to identify a step-length that archives adequate reductions in f at minimal cost.

Recently, Rivaie et al. [8] proposed a new-type conjugate gradient update parameter, i.e.,

$$\beta_k^{RMIL} = \frac{g_k^T y_{k-1}}{||d_{k-1}||^2}, \tag{1.3}$$

which is similar with β_k^{PRP}. It is clear that the corresponding RMIL method has the restart property. Unfortunately, the authors only proved its sufficient descent property and global convergence under the exact line search. Subsequently, Rivaie et al. [9] modified β_k^{RMIL} as:

$$\beta_k^{MRMIL} = \frac{g_k^T (g_k - g_{k-1} - d_{k-1})}{||d_{k-1}||^2}. \tag{1.4}$$

It is not difficult to find that β_k^{MRMIL} reduces to β_k^{RMIL} if the exact line search is used. They analyzed and discussed the sufficient decent property and global convergence of the MRMIL method under the strong Wolfe line search.

In recent years, some people are particularly interested in the three-term conjugate gradient methods. Based on the three-term form of the L-BFGS method [10], Zhang et al. [11] proposed a three-term PRP conjugate gradient (TTPRP) method, i.e.,

$$d_0 = -g_0, \quad d_k = -g_k + \beta_k^{PRP} + \theta_{k-1} y_{k-1},$$

where $\theta_k = -\frac{g_k^T d_{k-1}}{||g_{k-1}||^2}$. An attractive feature of the TTPRP method is that

$$d_k^T g_k = -||g_k||^2, \quad \forall k \geq 0, \tag{1.5}$$

holds without any line search. Under suitable conditions, this method is globally convergent when a modified Armijo line search is used. Subsequently, Zhang et al. [12] proposed a three-term HS conjugate gradient (TTHS) method, that is,

$$d_k = \begin{cases} -g_k, & if\ s_{k-1}^T y_{k-1} < \varepsilon_1 ||g_{k-1}||^r s_{k-1}^T s_{k-1}, \\ -g_k + \beta_k^{HS} d_{k-1} + \theta_{k-1} y_{k-1}, & otherwise \end{cases}$$

where $\theta_{k-1} = -\frac{g_k^T d_{k-1}}{d_{k-1}^T y_{k-1}}$, $s_{k-1} = x_k - x_{k-1}$, $r \geq 0$, $\varepsilon_1 > 0$. It is not difficult to prove that the search direction d_k also satisfies (1.5) independent of any line search. They proved the global convergence of the TTHS method under the standard Wolfe line search. Narushima et al. [13] constructed a family of three-term conjugate gradient methods, defined by

$$d_k = \begin{cases} -g_k, & if\ k = 0\ or\ g_k^T p_k = 0, \\ -g_k + \beta_k (g_k^T p_k)^\dagger [(g_k^T p_k) d_{k-1} - (g_k^T d_{k-1}) p_k], & otherwise \end{cases}$$

where p_k is a parameter vector. This is a general three-term conjugate gradient method which also always satisfies (1.5). The TTPRP and TTHS methods are the special cases of the above method. Interested readers may refer to more references about three-term conjugate gradient methods (Refs. [14–20]).

In this paper we are particularly interested in the structure of the TTPRP method [11]. We extend the RMIL method and the MRMIL method to establish two three-term conjugate gradient methods, which reduce to the RMIL method if the exact line search is implemented. An attractive feature of the proposed methods is that the search direction d_k always satisfies the sufficient descent condition, which is independent of any line search. Moreover, the global convergence properties of the proposed methods are established under the standard Wolfe line search.

The remainder of this paper is organized as follows. In Sect. 2 we propose the specific algorithm, and give some properties. In Sect. 3 we prove the global convergence of the proposed methods under suitable conditions, and the linear convergence rate is proved in Sect. 4. Finally, we provide numerical experiments to show their practical performance in Sect. 5.

2 Algorithm

In this section we describe the three-term RMIL and MRMIL methods whose form are similar to that of [11], but with different β_k and θ_k. The search direction d_k can be expressed as:

$$d_0 = -g_0, \quad d_k = -g_k + \beta_k d_{k-1} + \theta_k y_{k-1}, \tag{2.1}$$

where β_k is specified by (1.3) or (1.4), and

$$\theta_k = -\frac{g_k^T d_{k-1}}{||d_{k-1}||^2}. \tag{2.2}$$

It follows from (1.3) and (2.1), (2.2) that (1.5) holds for any $k \geq 0$, which is independent of any line search. On the other hand, from (1.4) and (2.1), (2.2) we have

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

Thus, the search direction d_k generated by the proposed methods always satisfies the sufficient descent condition ,i.e.,

$$g_k^T d_k \leq -c||g_k||^2, \quad \forall k \geq 0, \tag{2.3}$$

where $c = 1$. It is not difficult to find that the proposed methods reduce to the standard RMIL method if the exact line search is used. In order to prove the global convergence of the proposed methods, in this paper we consider that at the kth iteration the standard Wolfe line search is executed, that is the step-length α_k satisfying

$$f(x_k + \alpha_k d_k) - f(x_k) \le \rho \alpha_k g_k^T d_k, \tag{2.4}$$
$$g_{k+1}^T d_k \ge \sigma g_k^T d_k, \tag{2.5}$$

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

In the following, we describe the proposed three-term methods which we denote as TTRMIL method and TTMRMIL method, respectively.

Algorithm 2.1

Step 0: Set $\rho \in (0, 1)$, $\sigma \in (\rho, 1)$ *and* $\varepsilon > 0$, *and give the initial point* $x_0 \in R^n$. *Set* $k := 0$.

Step 1: If $||g_k|| \le \varepsilon$, *stop.*

Step 2: Determine α_k *by (2.4) and (2.5), set* $x_{k+1} = x_k + \alpha_k d_k$.

Step 3: Compute β_{k+1} *by (1.3) or (1.4), and obtain* θ_{k+1} *by (2.2).*

Step 4: Compute d_{k+1} *by (2.1).*

Step 5: Set $k := k + 1$, *go to step 1.*

Remark 2.1 From the previous analysis, in the proposed methods the search direction d_k satisfying (2.3) is a sufficient descent direction of f at point x_k. In addition, it follows from (2.3) that

$$||d_k|| \ge ||g_k||, \quad \forall k \ge 0. \tag{2.6}$$

This implies that the denominators of β_k^{RMIL}, β_k^{MRMIL} and θ_k are always greater than zero when $||g_k|| = 0$ is not achieved. Thus, the proposed methods are well defined.

3 Convergence analysis

In this section we need the following assumptions to analyze and prove the global convergence of the proposed methods.

Assumption 3.1 The level set $\Omega = \{x \in R^n | f(x) \le f(x_0)\}$ is bounded, i.e., there exists positive constant $B > 0$ such that $||x|| \le B$ for all $x \in \Omega$.

Assumption 3.2 In some neighborhood $\mathcal{C}$ of Ω, f is continuously differentiable and its gradient g is Lipschitz continuous, i.e., there exists a constant $L > 0$ such that

$$||g(x) - g(y)|| \le L||x - y||, \quad \forall x, y \in \mathcal{C}. \tag{3.1}$$

From (2.3) and (2.4), it is not difficult to find that the sequence $\{f(x_k)\}$ is decreasing. Thus, the sequence $\{x_k\}$ generated by the proposed methods is contained in Ω. Moreover, Assumptions 3.1 and 3.2 imply that there exists a constant $\gamma > 0$ such that

$$||g_k|| \le \gamma, \quad \forall x \in \Omega. \tag{3.2}$$

In the latter part of the paper, without specification, we always suppose that Assumptions 3.1 and 3.2 hold. At the same time we always assume that $g_k \neq 0$ holds for any $k \geq 0$. Although the search direction d_k generated by the proposed methods always satisfies (2.3), to prove the global convergence we need to constrain the choice of the step-length α_k. The following lemma shows that the standard Wolfe line search always gives a lower bound for the step-length α_k.

Lemma 3.1 *Let the sequences $\{g_k\}$ and $\{d_k\}$ be generated by the proposed methods, then we have*

$$\alpha_k \geq \frac{(1-\sigma)||g_k||^2}{L||d_k||^2}. \tag{3.3}$$

Proof From (2.5) we have

$$(\sigma - 1)g_k^T d_k \leq (g_{k+1} - g_k)^T d_k \leq ||g_{k+1} - g_k|| \cdot ||d_k|| \leq \alpha_k L||d_k||^2,$$

where the second inequality follows from Cauchy–Schwarz inequality, and the third inequality follows from (3.1). Since d_k satisfies (2.3) and $\sigma < 1$, it is clear that (3.3) holds.

To prove the global convergence of conjugate gradient methods, the Zoutendijk condition is usually used, which is first given by Wolfe [21] and Zoutendijk [22] with the standard Wolfe line search, respectively. The following lemma proves that for the proposed methods another form of the Zoutendijk condition also holds under the standard Wolfe line search. □

Lemma 3.2 *Let the sequences $\{g_k\}$ and $\{d_k\}$ be generated by the proposed methods, α_k is computed by the standard Wolfe line search, then we have*

$$\sum_{k=0}^{\infty} \frac{||g_k||^4}{||d_k||^2} < +\infty. \tag{3.4}$$

Proof From (2.4) we have

$$f(x_k) - f(x_{k+1}) \geq -\rho\alpha_k g_k^T d_k \geq \rho\alpha_k ||g_k||^2 \geq \frac{\rho(1-\sigma)||g_k||^4}{L||d_k||^2},$$

where the second inequality follows from (2.3), and the third inequality follows from (3.3). Therefore, it follows from Assumption 3.1 that we obtain (3.4).

As we know that the iteration of conjugate gradient methods may be fail, in the sense that $||g_k|| \geq \lambda$ for all $k \geq 0$, only if $||d_k|| \to \infty$ sufficiently rapidly. In other words, the sequence $\{||g_k||\}$ can be bounded away from zero only if $\sum_{k=0}^{\infty} \frac{1}{||d_k||} < +\infty$. In the following we proved a convergent result of the proposed methods with the standard Wolfe line search.

Theorem 3.1 *Let the sequences $\{g_k\}$ and $\{d_k\}$ be generated by the proposed methods, we have*

$$\lim_{k\to\infty} \inf ||g_k|| = 0. \tag{3.5}$$

Proof Suppose that (3.5) does not hold, i.e., there exists a constant $r > 0$ such that

$$||g_k|| > r, \quad \forall k \geq 0. \tag{3.6}$$

On the basis of the proposed two three-term conjugate gradient direction in the TTRMIL and TTMRMIL methods, the rest of the proof can be divided into two following cases:

Case 1 (TTRMIL) From (1.3) and (2.1), (2.2) we have

$$\begin{aligned}
||d_k|| &\leq ||g_k|| + |\beta_k^{RMIL}| \cdot ||d_{k-1}|| + |\theta_k| \cdot ||y_{k-1}|| \\
&\leq ||g_k|| + \frac{||g_k|| \cdot ||y_{k-1}||}{||d_{k-1}||^2} \cdot ||d_{k-1}|| + \frac{||g_k|| \cdot ||d_{k-1}||}{||d_{k-1}||^2} \cdot ||y_{k-1}|| \\
&\leq ||g_k|| + 2\frac{L||g_k|| \cdot ||x_k - x_{k-1}||}{||g_{k-1}||} \\
&\leq ||g_k|| + \frac{2L||g_k|| \cdot (||x_k|| + ||x_{k-1}||)}{||g_{k-1}||} \\
&\leq \gamma + \frac{4L\gamma B}{r} \triangleq \vartheta,
\end{aligned}$$

where the second inequality follows from Cauchy-Schwarz inequality, the third inequality follows from (2.6) and (3.1), the fourth inequality follows from Trigonometric inequality, and the final inequality follows from (3.2), (3.6) and Assumption 3.1.

Case 2 (TTMRMIL) From (1.4), it is easy to obtain that

$$|\beta_k^{MRMIL}| \leq |\beta_k^{RMIL}| + \frac{|g_k^T d_{k-1}|}{||d_{k-1}||^2} \leq |\beta_k^{RMIL}| + \frac{||g_k||}{||d_{k-1}||}, \tag{3.7}$$

where the second inequality obtains by Cauchy–Schwarz inequality.

Similar to what state in Case 1, for all $k \geq 0$, from (1.4) and (2.1), (2.2) we have

$$\begin{aligned}
||d_k|| &\leq ||g_k|| + |\beta_k^{MRMIL}| \cdot ||d_{k-1}|| + |\theta_k| \cdot ||y_{k-1}|| \\
&\leq 2||g_k|| + |\beta_k^{RMIL}| \cdot ||d_{k-1}|| + |\theta_k| \cdot ||y_{k-1}|| \\
&\leq 2\gamma + \frac{4L\gamma B}{r} \\
&= (\gamma + \vartheta),
\end{aligned}$$

where the second inequality obtains from(3.7), and the third inequality can be referred to the proof of Case 1. ◻

In summary, the sequence $\{||d_k||\}$ generated by the proposed methods has a common upper bound, i.e.

$$||d_k|| \leq M, \quad \forall k \geq 0, \tag{3.8}$$

where $M = \gamma + \vartheta$. Moreover, by using (3.6) and (3.8), it is clear that

$$\sum_{k=0}^{\infty} \frac{||g_k||^4}{||d_k||^2} \geq \sum_{k=0}^{\infty} \frac{r^4}{M^2} = +\infty, \tag{3.9}$$

which contradicts with (3.4). Thus, the result (3.5) holds.

4 Convergence rate

In this section we turn to prove the convergence rate of the proposed methods. The following assumption is also needed.

Assumption 4.1 Suppose that $f : R^n \to R$ is twice continuously differentiable, and that the sequence $\{x_k\}$ generated by the proposed methods converges to x^* at which $\nabla f(x^*) = 0$ and the Hessian matrix $\nabla^2 f(x^*)$ is positive definite.

From Assumption 4.1, there exists the neighborhood of x^* and constants $M \gg m > 0$ such that

$$m||p||^2 \leq p^T \nabla^2 f(x) p \leq M||p||^2, \forall x \in U(x^*), p \in R^n. \tag{4.1}$$

By using Taylor Theorem and (4.1), it is easy to obtain that

$$\frac{1}{2} m||x - x^*||^2 \leq f(x) - f(x^*) \leq \frac{1}{2} M||x - x^*||^2, \tag{4.2}$$

$$m||x - x^*|| \leq ||g(x)|| \leq M||x - x^*||. \tag{4.3}$$

Theorem 4.1 *Suppose that Assumptions 3.1, 3.2 and 4.1 hold, and that the sequence $\{x_k\}$ generated by the proposed methods converges to the unique solution x^*. Then, for the sufficient large k there exists a constant $a > 0$ such that*

$$||x_k - x^*|| \leq a\delta^k, \tag{4.4}$$

where $\delta \in (0, 1)$.

Proof From the proof of Theorem 3.1, for the proposed methods we always have

$$||d_k|| \leq 2||g_k|| + \frac{2L||g_k|| \cdot ||x_k - x_{k-1}||}{||g_{k-1}||} \leq \left(2 + \frac{4LB}{||g_{k-1}||}\right) ||g_k||, \tag{4.5}$$

where the final inequality follows from Assumption 3.1. Then from (2.3) to (2.4) we have

$$\begin{aligned} f(x_{k+1}) - f(x^*) &\leq f(x_k) - f(x^*) - \rho\alpha_k ||g_k||^2 \\ &\leq f(x_k) - f(x^*) - \frac{\rho(1-\sigma)||g_k||^4}{L||d_k||^2} \\ &\leq f(x_k) - f(x^*) - t_k ||g_k||^2 \end{aligned}$$

$$
\begin{aligned}
&\le f(x_k) - f(x^*) - t_k m^2 ||x_k - x^*||^2 \\
&\le f(x_k) - f(x^*) - \frac{2t_k m^2}{M}(f(x_k) - f(x^*)) \\
&= \delta^2 (f(x_k) - f(x^*)),
\end{aligned}
$$

where the second inequalities obtains from (3.3), the third inequalities uses (4.5) in which $t_k = \frac{\rho(1-\sigma)}{L\left(2+\frac{4LB}{||g_{k-1}||}\right)^2}$, the fourth inequality follows from (4.3), the the fifth inequality follows from (4.2) where $\delta^2 = 1 - \frac{2t_k m^2}{M}$.

From $L > 0$ and $0 < \rho < \sigma < 1$, we have $t_k > 0$. Since $M \gg m$, we always find appropriate M and m such that $\delta^2 \in (0, 1)$. Therefore, $\delta \in (0, 1)$ is defined well.

From the previous inequality, we have

$$
f(x_k) - f(x^*) \le \delta^2 (f(x_{k-1}) - f(x^*)) \le \cdots \le \delta^{2k} (f(x_0) - f(x^*)).
$$

This inequality together with (4.2) obtains that

$$
||x_k - x^*||^2 \le \frac{2}{m}(f(x_k) - f(x^*)) \le \frac{2}{m}\delta^{2k}(f(x_0) - f(x^*)),
$$

then we have

$$
||x_k - x^*|| \le a\delta^k,
$$

where $a = \frac{\sqrt{2(f(x_0) - f(x^*))}}{\sqrt{m}}$. This implies that the proposed methods are linear convergent. □

5 Preliminary numerical experiments

To give some insight into the behavior of the proposed methods, we compared their performances with those of TTPRP method [11] and MRMIL method [9]. The executed methods are coded in Fortran and compiled with f77 (default compiler settings) on a personal computer with Intel Core (TM) CPU 2.60 GHZ and 2.0 G memory. We selected a number of 27 large-scale unconstrained optimization problems in generalized or extended form from the Refs. [23] and [24]. For each problem we have taken five numerical experiments with the number of variables as $n = 1000, 6000, 11{,}000, 15{,}000, 20{,}000$. The executed methods implements the standard Wolfe line search with $\rho = 10^{-4}$ and $\sigma = 0.8$. For the terminating the executions, we used the same criterion $||g_k||_\infty \le 10^{-6}$, or the number of iteration exceeds 10,000, where $||\cdot||_\infty$ denotes the maximum absolute component of a vector. In addition, we also apply the same choice of α_k, i.e.

$$
\alpha_k = \begin{cases} 1, & k = 0, \\ \alpha_{k-1}\frac{||d_{k-1}||}{||d_k||}, & k \ge 1. \end{cases}
$$

Table 1 The numerical results obtained by the executed methods

No.	Dim	TTRMIL	TTMRMIL	TTPRP	MRMIL
Trigonometric	1000	30/0.03	32/0.03	31/0.03	31/0.02
	6000	33/0.19	34/0.19	33/0.17	33/0.17
	11,000	31/0.29	32/0.30	31/0.30	32/0.30
	15,000	34/0.43	35/0.46	34/0.43	37/0.56
	20,000	38/0.71	40/0.72	38/0.71	39/0.65
Extended Rosenbrock	1000	37/0.02	35/0.01	38/0.01	35/0.01
	6000	35/0.01	37/0.03	38/0.03	34/0.02
	11,000	37/0.05	39/0.05	40/0.07	39/0.06
	15,000	37/0.07	35/0.06	38/0.07	34/0.1
	20,000	37/0.1	36/0.11	38/0.11	36/0.09
Extended White	1000	34/0.01	35/0.02	33/0.01	35/0.01
	6000	35/0.03	37/0.03	33/0.04	35/0.03
	11,000	35/0.05	37/0.04	33/0.08	35/0.07
	15,000	35/0.08	37/0.08	33/0.08	36/0.08
	20,000	35/0.09	37/0.11	36/0.11	37/0.11
Extended Beale	1000	12/0.01	13/0.01	12/0.01	13/0.01
	6000	14/0.02	12/0.02	14/0.02	12/0.01
	11,000	12/0.01	12/0.01	12/0.01	12/0.02
	15,000	12/0.03	12/0.03	12/0.03	12/0.01
	20,000	12/0.03	13/0.04	12/0.03	12/0.03
Penalty	1000	10/0.01	10/0.01	10/0.01	10/0.01
	6000	10/0.02	18/0.01	10/0.01	10/0.01
	11,000	10/0.01	16/0.02	10/0.01	10/0.01
	15,000	11/0.03	36/0.12	11/0.04	10/0.02
	20,000	13/0.03	34/0.28	9/0.03	9/0.03
Generalized Tridiagonal 1	1000	29/0.02	23/0.01	25/0.02	65/0.03
	6000	25/0.03	24/0.03	23/0.01	61/0.19
	11,000	117/0.78	84/0.53	158/1.17	26/0.05
	15,000	31/0.09	77/0.64	23/0.08	142/1.39
	20,000	60/0.61	25/0.09	104/1.24	64/0.68
Extended Tridiagonal 1	1000	12/0.01	11/0.01	12/0.01	11/0.01
	6000	10/0.02	13/0.02	10/0.01	10/0.01
	11,000	10/0.01	11/0.01	10/0.02	10/0.02
	15,000	10/0.03	11/0.04	10/0.03	10/0.02
	20,000	10/0.03	11/0.01	10/0.01	9/0.01
Extended Three Expo Terms	1000	10/0.01	13/0.02	10/0.01	10/0.01
	6000	7/0.03	8/0.05	13/0.08	10/0.08
	11,000	8/0.11	9/0.14	8/0.11	9/0.11
	15,000	8/0.14	7/0.11	8/0.14	10/0.22
	20,000	7/0.22	8/0.2	7/0.17	16/0.35

Table 2 The numerical results by the executed methods

No.	Dim	TTRMIL	TTMRMIL	TTPRP	MRMIL
Generalized Tridiagonal 2	1000	68/0.01	63/0.01	64/0.01	72/0.01
	6000	75/0.07	71/0.07	59/0.05	72/0.08
	11,000	73/0.13	70/0.11	74/0.11	65/0.11
	15,000	61/0.16	70/0.16	65/0.15	54/0.12
	20,000	69/0.22	72/0.21	58/0.17	58/0.19
Generalized PSC1	1000	150/0.10	275/0.19	215/0.14	318/0.19
	6000	618/2.22	642/2.56	1271/3.90	593/2.34
	11,000	806/5.40	1108/6.92	762/4.49	787/4.14
	15,000	304/0.62	523/1.54	265/0.73	1707/12.39
	20,000	624/6.85	593/9.49	888/8.89	1042/9.61
Extended Powell	1000	49/0.02	45/0.01	53/0.01	249/0.03
	6000	67/0.09	65/0.06	70/0.05	1858/1.20
	11,000	69/0.0 5	92/0.08	61/0.08	120/0.17
	15,000	55/0.14	58/0.18	69/0.11	148/0.25
	20,000	71/0.19	99/0.24	75/0.17	135/0.33
Extended BD1	1000	52/0.02	47/0.02	52/0.01	52/0.01
	6000	55/0.11	55/0.11	55/0.11	55/0.11
	11,000	55/0.17	55/0.19	55/0.19	55/0.19
	15,000	54/0.36	54/0.25	54/0.25	54/0.25
	20,000	39/0.25	26/0.19	39/0.27	38/0.27
Extended Maratos	1000	68/0.01	68/0.01	66/0.02	76/0.02
	6000	69/0.05	69/0.06	65/0.04	Inf/Inf
	11,000	65/0.09	68/0.09	68/0.11	Inf/Inf
	15,000	66/0.12	69/0.12	66/0.12	Inf/Inf
	20,000	73/0.17	76/0.18	63/0.14	Inf/Inf
Extended Cliff	1000	28/0.02	26/0.02	19/0.01	9/0.02
	6000	13/0.06	12/0.06	9/0.05	Inf/Inf
	11,000	14/0.11	14/0.11	10/0.08	Inf/Inf
	15,000	13/0.16	14/0.14	11/0.11	Inf/Inf
	20,000	10/0.15	13/0.21	10/0.15	Inf/Inf
Quadratic Diagonal Perturbed	1000	167/0.03	244/0.05	169/0.03	299/0.03
	6000	459/0.69	677/0.54	405/0.31	687/0.55
	11,000	482/1.91	1024/1.52	514/0.74	959/1.41

In Tables 1, 2, 3 and 4 we report the numerical results obtained by using the executed methods to solve each test problem with different dimension sizes. The detailed results are presented in the form 'Niter/Time', where 'Niter' denotes the number of iterations, and 'Time' denotes the CPU time. If the number of iteration exceeds 10,000 before the algorithm terminates, we denote 'Niter/Time' as 'Inf/Inf'.

Table 3 The numerical results by the executed methods

No.	Dim	TTRMIL	TTMRMIL	TTPRP	MRMIL
	15,000	470/0.97	527/1.12	654/1.25	1413/2.8
	20,000	935/2.16	627/2.42	1000/2.6	1678/4.48
Extended Hiebert	1000	84/0.01	83/0.02	84/0.01	87/0.01
	6000	82/0.06	80/0.06	84/0.06	88/0.06
	11,000	83/0.15	85/0.12	83/0.11	87/0.11
	15,000	82/0.16	82/0.16	82/0.15	89/0.16
	20,000	80/0.2	86/0.22	83/0.21	83/0.21
Extended Quadratic Penalty QP1	1000	8/0.01	5460/3.16	3578/2.07	3904/2.24
	6000	45/0.13	30/0.07	Inf/Inf	5782/4.65
	11,000	6/0.01	6/0.02	6/0.01	6784/10.78
	15,000	20/0.11	Inf/Inf	11/0.04	8182/71.97
	20,000	6/0.03	Inf/Inf	Inf/Inf	Inf/Inf
Extended Tridiagonal 2	1000	104/0.06	36/0.04	85/0.07	36/0.01
	6000	66/0.18	96/0.33	139/0.56	106/0.34
	11,000	350/2.26	339/2.63	304/2.31	510/3.84
	15,000	597/6.87	607/6.44	565/5.89	235/2.31
	20,000	393/5.52	863/12.37	318/4.32	329/4.47
SINCOS	1000	40/0.01	41/0.01	33/0.01	32/0.01
	6000	40/0.03	38/0.03	32/0.01	43/0.03
	11,000	38/0.07	35/0.04	31/0.03	37/0.06
	15,000	40/0.08	35/0.08	32/0.06	40/0.08
	20,000	38/0.09	35/0.1	36/0.1	37/0.1
ARGLINB (CUTE)	1000	29/0.01	27/0.01	27/0.01	28/0.01
	6000	79/0.17	171/0.56	44/0.03	162/0.90
	11,000	48/0.06	65/0.07	54/0.07	130/0.85
	15,000	46/0.09	127/0.93	86/0.62	53/0.19
	20,000	217/2.58	194/2.38	202/2.4	215/2.53

In this section, we use the performance profiles of Dolan and Moré [25] to evaluate and compare the performances of the set of methods S on the set of test problems P, i.e., for n_s methods and n_s problems, the performance profile $\chi : R \rightarrow [0, 1]$ is defined as follows: for each $p \in P$ and for each $s \in S$, they defined $t_{p,s}$=computing time (similarly for the number of iterations) required to solve problems p by method s. The performance ratio is obtained by $\gamma_{p,s} = t_{p,s} / \min_{s \in S} t_{p,s}$. Then the performance profile is defined as $\chi_s(\tau) = \frac{1}{n_p} size\{p \in P | log_2(\gamma_{p,s}) \leq \tau\}$, where $\tau > 0$ and $sizeA$ denotes the number of the elements in the set A. The function χ_s is the distribution function for the performance ratio, and for a method χ_s is a nondecreasing, piecewise constant function, continuous from the right at each breakpoint. In addition, it is not difficult to find that $\chi(\tau)$ is the probability for method $s \in S$ such that $\log_2(\gamma_{p,s})$ is within a factor $\tau > 0$ of the best possible ratio. This means that a method with high

Table 4 The numerical results by the executed methods

No.	Dim	TTRMIL	TTMRMIL	TTPRP	MRMIL
NONDIA (CUTE)	1000	9/0.01	12/0.02	11/0.01	11/0.01
	6000	7/0.02	7/0.01	7/0.01	7/0.02
	11,000	7/0.01	7/0.01	7/0.01	7/0.01
	15,000	7/0.01	7/0.01	7/0.01	7/0.02
	20,000	10/0.03	7/0.02	7/0.02	7/0.01
DQDRTIC (CUTE)	1000	26/0.01	27/0.01	11/0.01	26/0.01
	6000	27/0.02	30/0.03	15/0.02	27/0.01
	11,000	31/0.05	31/0.03	10/0.02	31/0.03
	15,000	41/0.07	41/0.07	11/0.02	41/0.08
	20,000	20/0.05	20/0.05	11/0.03	21/0.05
Broyden Tridiagonal	1000	36/0.01	38/0.02	41/0.02	35/0.01
	6000	53/0.09	62/0.09	74/0.06	112/0.10
	11,000	51/0.1	52/0.1	62/0.09	105/0.15
	15,000	59/0.15	53/0.11	67/0.14	112/0.23
	20,000	50/0.14	116/0.35	52/0.14	128/0.38
EDENSCH (CUTE)	1000	158/0.12	88/0.06	86/0.07	31/0.01
	6000	88/0.36	150/0.72	54/0.17	74/0.29
	11,000	62/0.43	117/0.95	152/1.36	119/1.03
	15,000	56/0.44	98/0.10	96/0.01	79/0.75
	20,000	68/0.81	170/2.84	171/2.77	139/2.28
STAIRCASE S1	1000	17/0.01	20/0.01	18/0.01	21/0.01
	6000	14/0.02	17/0.02	15/0.01	23/0.03
	11,000	24/0.06	20/0.04	24/0.05	17/0.03
	15,000	17/0.03	19/0.05	15/0.04	21/0.06
	20,000	20/0.06	18/0.04	22/0.08	19/0.06
DIXON3DQ (CUTE)	1000	255/0.17	170/0.09	325/0.33	74/0.03
	6000	222/0.90	915/4.81	157/0.84	67/0.22
	11,000	827/6.57	508/3.91	644/5.53	305/2.20
	15,000	820/8.5	330/3.38	1001/10.41	645/6.76
	20,000	636/8.64	591/8.15	623/5.78	714/10.07
DENSCHNF (CUTE)	1000	132/0.11	172/0.16	169/0.12	Inf/Inf
	6000	243/1.06	311/1.88	283/1.13	1021/3.90
	11,000	735/6.12	771/6.59	688/5.17	Inf/Inf
	15,000	801/8.02	957/10.99	678/6.82	535/5.36
	20,000	567/7.7	793/11.01	599/8.15	875/12.06

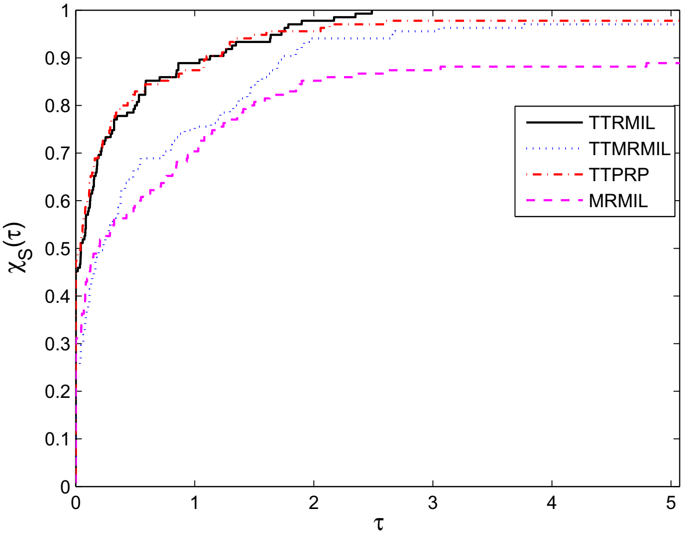

Fig. 1 Performance profiles of the executed methods with respect to the number of iterations

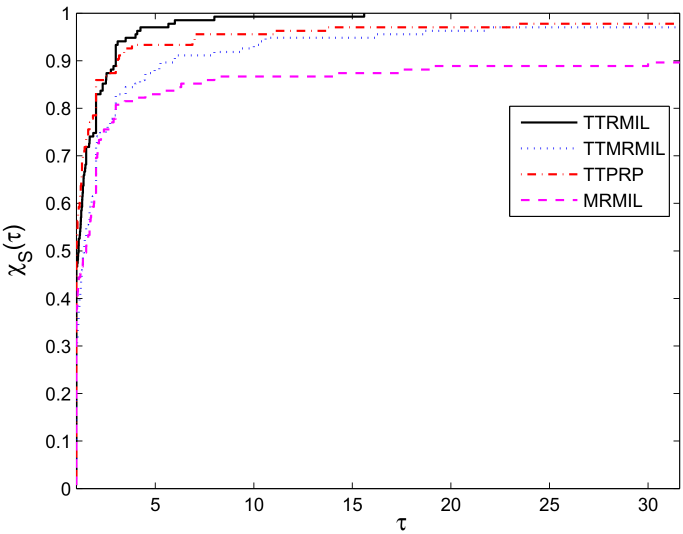

Fig. 2 Performance profiles of the executed methods with respect to CPU time

value of $\chi_s(\tau)$ is preferable or represent the best method when τ takes certain value. Figures 1 and 2, obtained by using $n_p = 135$, show the performances of the executed methods for the CPU time and the number of iterations.

From Tables 1, 2, 3 and 4, the TTRMIL method can solve all of the given test problems, the TTMRMIL method fails for Extended Quadratic Penalty QP1 with $n = 15{,}000$, $20{,}000$, the TTPRP method does not solve Extended Quadratic Penalty QP1 with $n = 6000$, $20{,}000$, and the RMIL method has more failures for the given problems. In this perspective, the TTRMIL method has the best performance. From the aspect of computing speed, Figs. 1 and 2 illustrates that the TTPRP method performs best, and the TTMRIL method is comparable with the TTPRP method. TTRMIL method is more effective than the TTMRMIL method. This may be that the TTRMIL method retains the restart property. Since the TTMRMIL method can generate the sufficient descent at each iteration, it performs better than the RMIL method.

References

1. Hestenes, M.R., Stiefel, E.L.: Methods of conjugate gradients for solving linear systems. J. Res. Natl. Bur. Stand. **5**, 409–432 (1952)
2. Fletcher, R., Reeves, C.M.: Function minimization by conjugate gradients. Comput. J. **7**, 149–154 (1964)
3. Polak, E., Ribire, G.: Note surla convergence des methodse de directions conjugees. Rev Francaise Imformmat Recherche Opertionelle **16**, 35–43 (1969)
4. Polyak, B.T.: The conjugate gradient method in extreme problems. USSR Comput. Math. Math. Phys. **9**, 94–112 (1969)
5. Fletcher, R.: Practical Methods of Optimization, 2nd edn. Wiley, New York (1987)
6. Liu, Y., Story, C.: Efficient generalized conjugate gradient algorithms. Part 1: theory. J. Optim. Theory Appl. **69**, 129–137 (1992)
7. Dai, Y.H., Yuan, Y.X.: Nonlinear conjugate gradient with a strong global convergence property. SIAM J. Optim. **10**, 177–182 (1999)
8. Rivaie, M., Mamat, M., June, L.W., Mohd, I.: A new class of nonlinear conjugate gradient coefficients with global convergence properties. Appl. Math. Comput. **218**, 11323–11332 (2012)
9. Rivaie, M., Mamat, M., Abashar, A.: A new class of nonlinear conjugate gradient coefficients with exact and inexact line searches. Appl. Math. Comput. **268**, 1152–1163 (2015)
10. Nocedal, J.: Updating quasi-Newton matrixes with limited storage. Math. Comput. **35**, 773–782 (1980)
11. 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**, 629–640 (2006)
12. Zhang, L., Zhou, W.J., Li, D.H.: Some descent three-term conjugate gradient methods and their global convergence. Optim. Methods. Softw. **22**, 697–711 (2007)
13. Narushima, Y., Yabe, H., Ford, J.A.: A three-term conjugate gradient method with sufficient descent property for unconstrained optimization. SIAM J. Optim. **21**, 212–230 (2011)
14. Liu, J.K., Li, S.J.: New three-term conjugate gradient method with guaranteed global convergence. Int. J. Comput. Math. **91**, 1744–1754 (2014)
15. Liu, J.K., Wu, X.S.: New three-term conjugate gradient method for solving unconstrained optimization problems. ScienceAsia **40**, 295–300 (2014)
16. Andrei, N.: A simple three-term conjugate gradient algorithm for unconstrained optimization. J. Comput. Appl. Math. **241**, 19–29 (2013)
17. Andrei, N.: On three-term conjugate gradient algorithm for unconstrained optimization. Appl. Math. Comput. **219**, 6316–6327 (2013)
18. Andrei, N.: An accelerated subspace minimization three-term conjugate gradient algorithm for unconstrained optimization. Numer. Algorithms **65**, 859–874 (2014)
19. Al-Bayati, A.Y., Sharif, W.H.: A new three-term conjugate gradient method for unconstrained optimization. Can. J. Sci. Eng. Math. **1**, 108–124 (2010)
20. Nazareth, L.: A conjugate direction algorithm without line search. J. Optim. Theory Appl. **23**, 373–387 (1977)
21. Wolfe, P.: Convergence conditions for ascent methods. SIAM Rev. **11**, 226–235 (1968)
22. Zoutendijk, G.: Nonlinear Programming, Computational Methods. In: Abadie, J. (ed.) Integer and Nonlinear Programming, pp. 38–86. North-Holland, Amsterdam (1970)

23. Bongartz, I., Conn, A.R., Gould, N.I.M., Toint., P.L.: CUTE: constrained and unconstrained testing environments. ACM Trans. Math. Softw. **21**, 123–160 (1995)
24. Andrei, N.: An unconstrained optimization test functions collection. Adv. Model. Optim. **10**, 147–161 (2008)
25. Dolan, E.D., Moré, J.J.: Benchmarking optimization software with performance profiles. Math. Program. **91**, 201–213 (2002)