Journal of Inequalities and Applications
a SpringerOpen Journal

RESEARCH Open Access

A new accelerated conjugate gradient method for large-scale unconstrained optimization

Yuting Chen[1,2], Mingyuan Cao[2] and Yueting Yang[2*]

*Correspondence:
yyt2858@163.com
[2]College of Mathematics and Statistics, Beihua University, Jilin, China
Full list of author information is available at the end of the article

Abstract

In this paper, we present a new conjugate gradient method using an acceleration scheme for solving large-scale unconstrained optimization. The generated search direction satisfies both the sufficient descent condition and the Dai–Liao conjugacy condition independent of line search. Moreover, the value of the parameter contains more useful information without adding more computational cost and storage requirements, which can improve the numerical performance. Under proper assumptions, the global convergence result of the proposed method with a Wolfe line search is established. Numerical experiments show that the given method is competitive for unconstrained optimization problems, with a maximum dimension of 100,000.

Keywords: Conjugate gradient; Descent condition; Dai–Liao conjugacy condition; Global convergence; Large-scale unconstrained optimization

1 Introduction

Consider the following unconstrained optimization problem:

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

where $f : \mathbb{R}^n \to \mathbb{R}$ is a continuously differentiable function, bounded below and its gradient is denoted by $g(x) = \nabla f(x)$. Conjugate gradient methods characterized by simplicity and low storage are efficient for solving (1), especially when the dimension n is large.

Starting from an initial guess $x_0 \in \mathbb{R}^n$, the conjugate gradient methods use the recurrence

$$x_{k+1} = x_k + \alpha_k d_k, \quad k \geq 0, \tag{2}$$

where x_{k+1} is the current iterate, $\alpha_k > 0$ is the step-length, which is obtained by some line search, and d_k is the search direction determined by

$$d_0 = -g_0, \qquad d_{k+1} = -g_{k+1} + \beta_k d_k, \quad k = 0, 1, 2, \ldots, \tag{3}$$

where $g_k = g(x_k)$. The scalar β_k is called the conjugate gradient parameter. There are many formulas to construct the scalar β_k, such as β_k^{PRP} [28, 29], β_k^{HS} [19] and β_k^{FR} [16].

The line search in conjugate gradient methods is usually based on the general Wolfe conditions [33, 34],

$$f(x_k + \alpha_k d_k) - f(x_k) \leq \rho \alpha_k g_k^{\mathrm{T}} d_k, \tag{4}$$

$$g_{k+1}^{\mathrm{T}} d_k \geq \sigma g_k^{\mathrm{T}} d_k, \tag{5}$$

where d_k is a descent direction and the constants ρ, σ satisfy $0 < \rho \leq \sigma \leq 1$. However, in order to establish the convergence and enhance the stability, the strong Wolfe conditions given by (4) and

$$\left| g_{k+1}^{\mathrm{T}} d_k \right| \leq \sigma \left| g_k^{\mathrm{T}} d_k \right| \tag{6}$$

are needed.

Recently efforts have been made to modify conjugate gradient methods for minimizing unconstrained optimization. In a natural way, Dai and Liao [8] extended the classical conjugate condition $y_k^{\mathrm{T}} d_{k+1} = 0$ to

$$d_{k+1}^{\mathrm{T}} y_k = -t g_{k+1}^{\mathrm{T}} s_k, \tag{7}$$

where $s_k = x_{k+1} - x_k$, $y_k = g_{k+1} - g_k$ and t is a positive parameter. Based on the Dai–Liao conjugacy condition (7), [8] introduced the conjugate gradient parameter β_k^{DL} as follows:

$$\beta_k^{\mathrm{DL}} = \frac{g_{k+1}^{\mathrm{T}} y_k}{d_k^{\mathrm{T}} y_k} - t \frac{g_{k+1}^{\mathrm{T}} s_k}{d_k^{\mathrm{T}} y_k}. \tag{8}$$

Having applied modified secant equations, many researchers have derived various conjugate gradient methods [7, 17, 21, 22, 25, 26, 30, 35, 44]. Moreover, combining with a quasi-Newton updating technique, conjugate gradient methods can be considered as a special type of quasi-Newton methods. From (3), (7) and Perry's point of view in [27], we can rewrite the search direction as follows:

$$d_{k+1} = -Q_{k+1} g_{k+1}, \tag{9}$$

where

$$Q_{k+1} = I - \frac{s_k y_k^{\mathrm{T}}}{s_k^{\mathrm{T}} y_k} + t \frac{s_k s_k^{\mathrm{T}}}{s_k^{\mathrm{T}} y_k}. \tag{10}$$

Obviously, the Dai–Liao method can be considered as a special type of quasi-Newton method in which the matrix Q_{k+1} is used to approximate the inverse Hessian of the objective function. Since the matrix Q_{k+1} is nonsymmetric and does not satisfy the secant condition, (9) cannot be regarded as a quasi-Newton direction from a strict point of view.

To overcome the above shortcomings and improve the numerical performance of conjugate gradient methods, Andrei [1, 2] proposed the following matrix Q_{k+1}^{A}:

$$Q_{k+1}^{\mathrm{A}} = I - \frac{s_k y_k^{\mathrm{T}} - y_k s_k^{\mathrm{T}}}{y_k^{\mathrm{T}} s_k} + t \frac{s_k s_k^{\mathrm{T}}}{y_k^{\mathrm{T}} s_k} \tag{11}$$

to replace the matrix Q_{k+1} in (10). The parameter t in the last term on the right-hand side is calculated with $t = 1 + \frac{\|y_k\|^2}{y_k^{\mathrm{T}} s_k}$ and $t = 1 + 2\frac{\|y_k\|^2}{y_k^{\mathrm{T}} s_k}$, corresponding to the THREECG method [1] and the TTCG method [2], respectively. The search directions satisfy not only the descent condition but also the conjugacy condition, independent of the line search. Both of the methods can be regarded as modifications of the classical HS or of the CG_DESCENT conjugate gradient methods. Numerical results support this claim.

Motivated by [1] and [2], Deng and Wan [14] presented a symmetric matrix to estimate the inverse Hessian approximation as follows:

$$Q_{k+1}^{\mathrm{B}} = I - \frac{y_k s_k^{\mathrm{T}} + y_k s_k^{\mathrm{T}}}{y_k^{\mathrm{T}} s_k} + t\frac{s_k s_k^{\mathrm{T}}}{y_k^{\mathrm{T}} s_k}, \tag{12}$$

where $t = 1 - \frac{\|y_k\|^2}{y_k^{\mathrm{T}} s_k}$. The search direction in this method (MTHREECG) is close to the Newton direction and satisfies the Dai–Liao conjugacy condition. Then they restricted $t = 1 - \min\{1, \frac{\|y_k\|^2}{y_k^{\mathrm{T}} s_k}\}$ and obtained the descent property. Numerical results show that the MTHREECG method outperforms the THREECG method and the CG_DESCENT method.

More recently, Yao and Ning [37] suggested the following symmetric matrix:

$$Q_{k+1}^{\mathrm{C}} = I - t_k\frac{y_k s_k^{\mathrm{T}} + y_k s_k^{\mathrm{T}}}{y_k^{\mathrm{T}} s_k} + \frac{s_k s_k^{\mathrm{T}}}{y_k^{\mathrm{T}} s_k}, \tag{13}$$

where the positive parameter t_k is determined by minimizing the distance of Q_{k+1}^{C} and the self-scaling memoryless BFGS matrix in the Frobenius norm. In this method (NTAP), they let $a_k = \frac{\|s_k\|^2\|y_k\|^2}{(s_k^{\mathrm{T}} y_k)^2}$, then t_k can be expressed as $t_k = \min\{\frac{1}{1+a_k}, \frac{s_k^{\mathrm{T}} y_k}{\|y_k\|^2}\}$. The sufficient descent property of the search direction depends neither on the line search, nor on the convexity of objective function. For relevant research see [3–6, 10–13, 20, 23, 24, 31, 32, 36, 38–43].

By focusing on the above research, we are interested in developing a new accelerated conjugate gradient method (NACG) for large-scale unconstrained optimization. The generated search direction satisfies both sufficient descent condition and Dai–Liao conjugacy condition. The parameter in the given method provides more useful information and adds no extra computational and storage burden. In addition, the proposed method has an obvious improvement in computational performance, especially in dealing with large-scale unconstrained optimization problems.

The rest of this paper is organized as follows. In the next section, we will describe the framework of the new method and the choice of parameter in generated search direction. Global convergence results of the obtained method will be established under appropriate conditions in Sect. 3. Section 4 is devoted to numerical experiments and comparisons with some other efficient conjugate gradient algorithms for solving unconstrained optimization problems with different dimensions. Conclusions are drawn in Sect. 5.

2 The NACG method

In this section, we state our new accelerated conjugate gradient method exploiting BFGS updating technology, for which at each step both the sufficient descent condition and the Dai–Liao conjugacy condition are satisfied, independent of the line search.

It is well known that the BFGS method is one of the most efficient quasi-Newton methods. By introducing two adaptive parameters for adjusting, we give the following matrix:

$$Q_{k+1} = I - t_{k_1}\frac{s_k y_k^{\mathrm{T}} + y_k s_k^{\mathrm{T}}}{y_k^{\mathrm{T}} s_k} + t_{k_2}\frac{s_k s_k^{\mathrm{T}}}{y_k^{\mathrm{T}} s_k}, \tag{14}$$

where the parameters t_{k_1} and t_{k_2} are determined in the following.

In a sense, the method of form (9) and (14) could be considered as a self-adaptive memoryless BFGS method. Substituting (14) into (9), we get

$$\begin{aligned} d_{k+1} &= -\left(I - t_{k_1}\frac{s_k y_k^{\mathrm{T}} + y_k s_k^{\mathrm{T}}}{y_k^{\mathrm{T}} s_k} + t_{k_2}\frac{s_k s_k^{\mathrm{T}}}{y_k^{\mathrm{T}} s_k}\right) g_{k+1} \\ &= -g_{k+1} + \left(t_{k_1}\frac{y_k^{\mathrm{T}} g_{k+1}}{y_k^{\mathrm{T}} s_k} - t_{k_2}\frac{s_k^{\mathrm{T}} g_{k+1}}{y_k^{\mathrm{T}} s_k}\right)\cdot s_k + t_{k_1}\frac{s_k^{\mathrm{T}} g_{k+1}}{y_k^{\mathrm{T}} s_k}\cdot y_k. \end{aligned} \tag{15}$$

Let

$$a_k = t_{k_1}\frac{y_k^{\mathrm{T}} g_{k+1}}{y_k^{\mathrm{T}} s_k} - t_{k_2}\frac{s_k^{\mathrm{T}} g_{k+1}}{y_k^{\mathrm{T}} s_k}, \tag{16}$$

$$b_k = t_{k_1}\frac{s_k^{\mathrm{T}} g_{k+1}}{y_k^{\mathrm{T}} s_k}, \tag{17}$$

then (15) takes the form of three-term conjugate gradient,

$$d_{k+1} = -g_{k+1} + a_k s_k + b_k y_k. \tag{18}$$

In what follows, we discuss the choices for the two parameters t_{k_1} and t_{k_2}. The parameters are selected in such a manner that the Dai–Liao conjugacy condition and the sufficient descent condition are satisfied from iteration to iteration.

Consider the Dai–Liao conjugacy condition (7) with $t = 1$, i.e.,

$$y_k^{\mathrm{T}} d_{k+1} = -s_k^{\mathrm{T}} g_{k+1}. \tag{19}$$

Substituting (15) into (19), by simple calculation, we obtain

$$-y_k^{\mathrm{T}} g_{k+1} + t_{k_1} y_k^{\mathrm{T}} g_{k+1} - t_{k_2} s_k^{\mathrm{T}} g_{k+1} + t_{k_1}\frac{s_k^{\mathrm{T}} g_{k+1}}{y_k^{\mathrm{T}} s_k}\cdot y_k^{\mathrm{T}} y_k = -s_k^{\mathrm{T}} g_{k+1},$$

which yields

$$t_{k_2} = t_{k_1}\frac{y_k^{\mathrm{T}} y_k}{y_k^{\mathrm{T}} s_k} + t_{k_1}\frac{y_k^{\mathrm{T}} g_{k+1}}{s_k^{\mathrm{T}} g_{k+1}} - \frac{y_k^{\mathrm{T}} g_{k+1}}{s_k^{\mathrm{T}} g_{k+1}} + 1. \tag{20}$$

Since the descent property of the search direction, $g_{k+1}^{\mathrm{T}} d_{k+1} < 0$, is crucial for the convergence analysis. It is easy to see

$$\begin{aligned} g_{k+1}^{\mathrm{T}} d_{k+1} &= -\|g_{k+1}\|^2 + 2t_{k_1} \frac{y_k^{\mathrm{T}} g_{k+1} \cdot g_{k+1}^{\mathrm{T}} s_k}{y_k^{\mathrm{T}} s_k} - t_{k_2} \frac{(g_{k+1}^{\mathrm{T}} s_k)^2}{y_k^{\mathrm{T}} s_k} \\ &\leq -\|g_{k+1}\|^2 + 2t_{k_1} |g_{k+1}| \left| \frac{y_k g_{k+1}^{\mathrm{T}} s_k}{y_k^{\mathrm{T}} s_k} \right| - t_{k_2} \frac{(g_{k+1}^{\mathrm{T}} s_k)^2}{y_k^{\mathrm{T}} s_k} \\ &\leq -\|g_{k+1}\|^2 + t_{k_1} \left[\|g_{k+1}\|^2 + \frac{(g_{k+1}^{\mathrm{T}} s_k)^2}{(y_k^{\mathrm{T}} s_k)^2} \cdot y_k^{\mathrm{T}} y_k \right] - t_{k_2} \frac{(g_{k+1}^{\mathrm{T}} s_k)^2}{y_k^{\mathrm{T}} s_k} \\ &= -(1 - t_{k_1}) \|g_{k+1}\|^2 - \left(\frac{g_{k+1}^{\mathrm{T}} s_k}{y_k^{\mathrm{T}} s_k} \right)^2 \left(t_{k_2} y_k^{\mathrm{T}} s_k - t_{k_1} y_k^{\mathrm{T}} y_k \right). \end{aligned}$$

If we restrict $|t_{k_1}| < 1$ and take

$$t_{k_2} = t_{k_1} \frac{y_k^{\mathrm{T}} y_k}{y_k^{\mathrm{T}} s_k}, \tag{21}$$

then $g_{k+1}^{\mathrm{T}} d_{k+1} < 0$ holds. Substituting (21) into (20), we have

$$t_{k_1} = 1 - \frac{s_k^{\mathrm{T}} g_{k+1}}{y_k^{\mathrm{T}} g_{k+1}}. \tag{22}$$

If $|1 - \frac{s_k^{\mathrm{T}} g_{k+1}}{y_k^{\mathrm{T}} g_{k+1}}| \geq 1$, we set $t_{k_1} = 0$. This implies a restarted scheme. Taking into consideration the acceleration technique, the new accelerated conjugate gradient method (NACG) can be suggested.

Algorithm 1 (NACG)

Step 0. Choose an initial point $x_0 \in \mathbb{R}^n$, $\varepsilon > 0$, and compute $f_0 = f(x_0)$, $g_0 = \nabla f(x_0)$. Set $d_0 := -g_0$ and $k := 0$.

Step 1. If $\|g_k\| < \varepsilon$, stop, else go to Step 2.

Step 2. Compute a step-length α_k by Wolfe line search (4) and (5).

Step 3. Compute x_{k+1} by the acceleration scheme,

3.1. Compute $z = x_k + \alpha_k d_k$, $g_z = \nabla f(z)$ and $y_z = g_k - g_z$;

3.2. Compute $\bar{a}_k = \alpha_k g_k^{\mathrm{T}} d_k$ and $\bar{b}_k = -\alpha_k y_k^{\mathrm{T}} d_k$;

3.3. Acceleration scheme. If $\bar{b}_k > 0$, then compute $\xi_k = -\bar{a}_k / \bar{b}_k$ and update the variables as $x_{k+1} = x_k + \xi_k \alpha_k d_k$, otherwise update the variables as $x_{k+1} = x_k + \alpha_k d_k$.

Step 4. Compute $f_{k+1} = f(x_{k+1})$, $g_{k+1} = g(x_{k+1})$, $s_k = x_{k+1} - x_k$ and $y_k = g_{k+1} - g_k$.

Step 5. Compute $s_k^{\mathrm{T}} g_{k+1}$, $y_k^{\mathrm{T}} g_{k+1}$, $y_k^{\mathrm{T}} s_k$ and $y_k^{\mathrm{T}} y_k$, respectively.

Step 6. Compute t_{k_1} and t_{k_2} by

$$t_{k_1} = \begin{cases} 1 - \frac{s_k^{\mathrm{T}} g_{k+1}}{y_k^{\mathrm{T}} g_{k+1}}, & \text{if } 0 < \frac{s_k^{\mathrm{T}} g_{k+1}}{y_k^{\mathrm{T}} g_{k+1}} < 2, \\ 0, & \text{else}, \end{cases} \tag{23}$$

and

$$t_{k_2} = t_{k_1} \frac{y_k^{\mathrm{T}} y_k}{y_k^{\mathrm{T}} s_k}, \tag{24}$$

respectively.

Step 7. Compute a_k and b_k by (16) and (17), respectively.

Step 8. Set $d_{k+1} = -g_{k+1} + a_k s_k + b_k y_k$. Set $k := k + 1$ and go to Step 1.

In Algorithm 1, Step 3 corresponds to the acceleration scheme. In Step 6, the parameter t_{k_1} defined by (23) satisfies $|t_{k_1}| < 1$, and the parameter t_{k_2} could be determined by the equality with t_{k_1}. Furthermore, the main computational cost lies in $s_k^{\mathrm{T}} g_{k+1}$, $y_k^{\mathrm{T}} g_{k+1}$, $y_k^{\mathrm{T}} s_k$ and $y_k^{\mathrm{T}} y_k$ in Step 5. It costs $O(4n)$ operations to compute the values of t_{k_1} and t_{k_2}, and further get the values of a_k and b_k. No additional storage cost is required during the calculation. Compared with the existing effective algorithms TTCG [2], MTHREECG [14] and NTAP [37], the TTCG and the MTHREECG require $O(4n)$ operations, while the NTAP requires $O(5n)$ operations. In one word, our algorithm NACG is competitive in computational cost.

The sufficient descent condition and Dai–Liao conjugacy condition of the generated search direction holds independent of line search, a concept we discuss next.

Lemma 2.1 *Suppose that the search direction d_{k+1} is generated by Algorithm 1. Then d_{k+1} shows sufficient descent, i.e., $g_{k+1}^{\mathrm{T}} d_{k+1} \le -c\|g_{k+1}\|^2$, where the constant $c > 0$.*

Proof From (16)–(18), we have

$$\begin{aligned} g_{k+1}^{\mathrm{T}} d_{k+1} &= -\|g_{k+1}\|^2 + \left(t_{k_1} \frac{y_k^{\mathrm{T}} g_{k+1}}{y_k^{\mathrm{T}} s_k} - t_{k_2} \frac{s_k^{\mathrm{T}} g_{k+1}}{y_k^{\mathrm{T}} s_k} \right) \cdot g_{k+1}^{\mathrm{T}} s_k + t_{k_1} \frac{s_k^{\mathrm{T}} g_{k+1}}{y_k^{\mathrm{T}} s_k} \cdot g_{k+1}^{\mathrm{T}} y_k \\ &\le -(1 - t_{k_1})\|g_{k+1}\|^2 - \left(\frac{g_{k+1}^{\mathrm{T}} s_k}{y_k^{\mathrm{T}} s_k} \right)^2 \left(t_{k_2} y_k^{\mathrm{T}} s_k - t_{k_1} y_k^{\mathrm{T}} y_k \right). \end{aligned}$$

Combining with (23) and (24), it follows that $g_{k+1}^{\mathrm{T}} d_{k+1} \le -c\|g_{k+1}\|^2$ with $c := 1 - t_{k_1} > 0$. The proof is completed. □

Lemma 2.2 *Suppose that the search direction d_{k+1} is generated by Algorithm 1. Then d_{k+1} satisfies the Dai–Liao conjugacy condition* (19).

Proof From (16)–(18), we have

$$\begin{aligned} y_k^{\mathrm{T}} d_{k+1} &= -y_k^{\mathrm{T}} g_{k+1} + \left(t_{k_1} \frac{y_k^{\mathrm{T}} g_{k+1}}{y_k^{\mathrm{T}} s_k} - t_{k_2} \frac{s_k^{\mathrm{T}} g_{k+1}}{y_k^{\mathrm{T}} s_k} \right) \cdot y_k^{\mathrm{T}} s_k + t_{k_1} \frac{s_k^{\mathrm{T}} g_{k+1}}{y_k^{\mathrm{T}} s_k} \cdot y_k^{\mathrm{T}} y_k \\ &= -y_k^{\mathrm{T}} g_{k+1} + t_{k_1} y_k^{\mathrm{T}} g_{k+1} - t_{k_2} s_k^{\mathrm{T}} g_{k+1} + t_{k_1} \frac{s_k^{\mathrm{T}} g_{k+1}}{y_k^{\mathrm{T}} s_k} \cdot y_k^{\mathrm{T}} y_k. \end{aligned}$$

Substituting (23) and (24) into the above equality yields

$$\begin{aligned} y_k^{\mathrm{T}} d_{k+1} &= -y_k^{\mathrm{T}} g_{k+1} + \left(1 - \frac{s_k^{\mathrm{T}} g_{k+1}}{y_k^{\mathrm{T}} g_{k+1}} \right) \cdot y_k^{\mathrm{T}} g_{k+1} - t_{k_1} \frac{y_k^{\mathrm{T}} y_k}{y_k^{\mathrm{T}} s_k} \cdot s_k^{\mathrm{T}} g_{k+1} \\ &\quad + t_{k_1} \frac{s_k^{\mathrm{T}} g_{k+1}}{y_k^{\mathrm{T}} s_k} \cdot y_k^{\mathrm{T}} y_k \end{aligned}$$

$$= -y_k^{\mathrm{T}} g_{k+1} + y_k^{\mathrm{T}} g_{k+1} - s_k^{\mathrm{T}} g_{k+1}$$
$$= -s_k^{\mathrm{T}} g_{k+1},$$

which completes the proof. □

3 Convergence analysis

In this section, under appropriate assumptions, the global convergence of Algorithm 1 is established. Without loss of generality, we make the following basic assumptions.

Assumption (i) The level set

$$\Omega = \left\{ x \in \mathbb{R}^n : f(x) \leq f(x_0) \right\} \tag{25}$$

is bounded, i.e., there exists a constant $B > 0$ such that

$$\|x\| \leq B, \quad \forall x \in \Omega. \tag{26}$$

Assumption (ii) The function $f : \mathbb{R}^n \to \mathbb{R}$ is continuously differentiable and its gradient is Lipschitz continuous in a neighborhood $\mathbb{N}$ of Ω, i.e., there exists a constant $L > 0$ such that

$$\left\| g(x) - g(y) \right\| \leq L \|x - y\|, \quad \forall x, y \in \mathbb{N}. \tag{27}$$

Under the above assumptions, we can easily see that there exists a constant $\Gamma > 0$ such that

$$\left\| g(x) \right\| \leq \Gamma, \quad \forall x \in \Omega. \tag{28}$$

Although the search direction d_{k+1} generated by Algorithm 1 is always a descent direction, in order to obtain the convergence of Algorithm 1, we need to derive a lower bound for the step-length α_k.

Lemma 3.1 *Suppose that Assumption (ii) holds and $\{d_k\}$ is generated by Algorithm 1. Then the step-length α_k satisfies*

$$\alpha_k \geq \frac{(\sigma - 1) g_k^{\mathrm{T}} d_k}{L \|d_k\|^2}. \tag{29}$$

The following lemma is called the Zoutendijk condition [45], which is often used to prove global convergence of conjugate gradient methods.

Lemma 3.2 *Suppose that the assumptions hold. Consider the algorithm (2) and (18), where d_k is a descent direction and α_k is obtained by a Wolfe line search (4) and (5). Then*

$$\sum_{k \geq 0} \frac{(g_k^{\mathrm{T}} d_k)^2}{\|d_k\|^2} < \infty. \tag{30}$$

The next lemma shows the sequence of gradient norms $\|g_k\|$ is bounded away from zero only if $\sum_{k\geq 0} 1/\|d_k\| < +\infty$ for any conjugate gradient methods with strong Wolfe line search (4) and (6).

Lemma 3.3 *Suppose that the assumptions hold. Consider the algorithm* (2) *and* (18)*, where* d_k *is a descent direction and* α_k *is obtained by a strong Wolfe line search* (4) *and* (6)*. If*

$$\sum_{k\geq 0} \frac{1}{\|d_k\|^2} = +\infty, \tag{31}$$

then

$$\liminf_{k\to\infty} \|g_k\| = 0. \tag{32}$$

The proofs of Lemmas 3.1–3.3 refer to [1, 2], which are omitted here.

For uniformly convex functions, we establish the following global convergence result of Algorithm 1.

Theorem 3.1 *Suppose that the assumptions hold. Let* $\{x_k\}$ *and* $\{d_k\}$ *be generated by Algorithm 1. If* f *is a uniformly convex function on* Ω*, i.e., there exists a constant* $\mu > 0$ *such that*

$$\big(\nabla f(x) - \nabla f(y)\big)^{\mathrm{T}}(x-y) \geq \mu\|x-y\|^2, \quad \forall x, y \in \mathbb{N}, \tag{33}$$

then

$$\lim_{k\to\infty} \|g_k\| = 0. \tag{34}$$

Proof From (27), it follows that

$$\|y_k\| \leq L\|s_k\|. \tag{35}$$

From (33), we have

$$y_k^{\mathrm{T}} s_k \geq \mu\|s_k\|^2. \tag{36}$$

By the use of the Cauchy inequality and (36), it is obvious that $\mu\|s_k\|^2 \leq y_k^{\mathrm{T}} s_k \leq \|y_k\|\|s_k\|$, i.e.,

$$\mu\|s_k\| \leq \|y_k\|. \tag{37}$$

We get from (23) and (24)

$$\begin{aligned} |t_{k_2}| &\leq |t_{k_1}| \frac{|y_k^{\mathrm{T}} y_k|}{|y_k^{\mathrm{T}} s_k|} \\ &\leq \frac{L^2\|s_k\|^2}{\mu\|s_k\|^2} \\ &= \frac{L^2}{\mu} \triangleq M_0. \end{aligned} \tag{38}$$

On the other hand, from the definition of a_k and b_k in (16) and (17), we obtain

$$\begin{aligned}|a_k| &\le |t_{k_1}| \cdot \frac{|y_k^{\mathrm{T}} g_{k+1}|}{|y_k^{\mathrm{T}} s_k|} + |t_{k_2}| \frac{|s_k^{\mathrm{T}} g_{k+1}|}{|y_k^{\mathrm{T}} s_k|} \\ &\le \frac{\Gamma L}{\mu} \cdot \frac{1}{\|s_k\|} + \frac{M_0 \Gamma}{\mu} \cdot \frac{1}{\|s_k\|} \\ &= \frac{\Gamma L + M_0 \Gamma}{\mu} \cdot \frac{1}{\|s_k\|} \triangleq M_1 \frac{1}{\|s_k\|} \end{aligned} \tag{39}$$

and

$$\begin{aligned}|b_k| &\le |t_{k_1}| \cdot \frac{|s_k^{\mathrm{T}} g_{k+1}|}{|y_k^{\mathrm{T}} s_k|} \\ &\le \frac{\Gamma \|s_k\|}{\mu \|s_k\|^2} \\ &\le \frac{\Gamma L}{\mu} \cdot \frac{1}{\|y_k\|} \triangleq M_2 \frac{1}{\|y_k\|}. \end{aligned} \tag{40}$$

Therefore, using (39) and (40) in (18), we get

$$\begin{aligned}\|d_{k+1}\| &\le \|g_{k+1}\| + |a_k| \|s_k\| + |b_k| \|y_k\| \\ &\le \Gamma + M_1 + M_2 \triangleq M, \end{aligned} \tag{41}$$

showing that (31) holds. From Lemma 3.3, it follows that (32) is true, which for uniformly convex functions is equivalent to (34). The proof is completed. □

4 Numerical results

In this section, we report the numerical results for some unconstrained problems from [9] to show the efficiency of Algorithm 1 (NACG). All codes are written in Matlab R2013a and ran on PC with 1.80 GHz CPU processor and 8.00 GB RAM memory.

We compare NACG against TTCG [2], MTHREECG [14] and NTAP [37], which have a similar structure in search direction and have been reported to be superior to the classical PRP method, HS method and CG-DESCENT [18] method, etc.

The iteration is terminated by the following condition:

$$\|g_k\| \le \varepsilon \quad \text{or} \quad \left|f(x_{k+1}) - f(x_k)\right| \le \varepsilon \max\left\{1.0, \left|f(x_k)\right|\right\}. \tag{42}$$

All algorithms have the same stopping criteria. We set the parameters as $\varepsilon = 10^{-6}$ in (42), and $\rho = 0.0001$, $\sigma = 0.8$ in a Wolfe line search (4) and (5). The other parameters are set as default. Table 1 lists the test problems and their dimensions.

According to a comparison of four algorithms for the 300 test problems with different dimensions, we can see that there is only one problem that the NACG and the MTHREECG cannot solve, while the TTCG does 98 percent of problems and the NTAP does 84.2 percent of problems, respectively.

We employ the profiles by Dolan and Moré [15] to analyze the efficiency of the NACG. In a performance profile plot, the horizontal axis gives the percentage (τ) of the test problems

Table 1 The test problems and their dimensions

No.	Prob	dim
1.	*Penalty function* II	500, ..., 900, 1000
2.	*Chebyquad function*	500, ..., 900, 1000, 2000, ..., 5000
3.	*Nearly separable function*	500, ..., 900, 1000, 2000, ..., 5000
4.	*Integral equation function*	500, ..., 900, 1000, 2000, ..., 5000
5.	*Penalty function* I	500, ..., 900, 1000, ..., 9000, 10,000, ..., 90,000, 100,000
6.	*Extended Powell singular function*	500, ..., 900, 1000, ..., 9000, 10,000, ..., 90,000, 100,000
7.	*Variable dimension function*	500, ..., 900, 1000, ..., 9000, 10,000, ..., 90,000, 100,000
8.	*Schittkowski function* 302	500, ..., 900, 1000, ..., 9000, 10,000, ..., 90,000, 100,000
9.	*Generalized Rosebrock function*	500, ..., 900, 1000, ..., 9000, 10,000, ..., 90,000, 100,000
10.	*Extended Rosenbrock function*	500, ..., 900, 1000, ..., 9000, 10,000, ..., 90,000, 100,000
11.	*Boundary value function*	500, ..., 900, 1000, ..., 9000, 10,000, ..., 90,000, 100,000
12.	*Broyden tridiagonal function*	500, ..., 900, 1000, ..., 9000, 10,000, ..., 90,000, 100,000
13.	*Separable cubic function*	500, ..., 900, 1000, ..., 9000, 10,000, ..., 90,000, 100,000
14.	*Yang tridiagonal function*	500, ..., 900, 1000, ..., 9000, 10,000, ..., 90,000, 100,000
15.	*Allgower function*	500, ..., 900, 1000, ..., 9000, 10,000, ..., 90,000, 100,000

Figure 1 The number of iterations

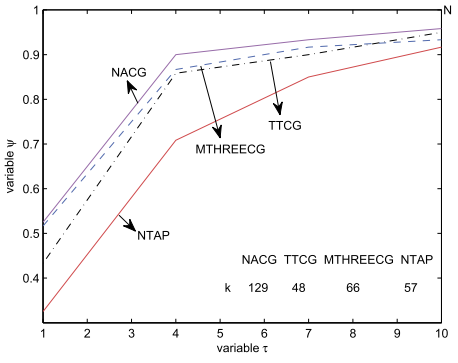
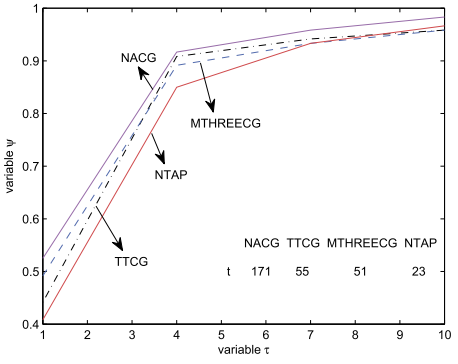


Figure 2 The CPU time



for which a method is the fastest (efficiency), while the vertical side gives the percentage (ψ) of the test problems that are successfully solved by each of the methods. Consequently, the top curve is the method that solved the most problems in a time that is within a factor of the best time.

Figures 1–4 plot the performance profiles for the number of iterations (k), the CPU time (t), the number of function evaluations (nf) and the number of gradient evaluations (ng), respectively.

Figure 3 The number of function evaluations

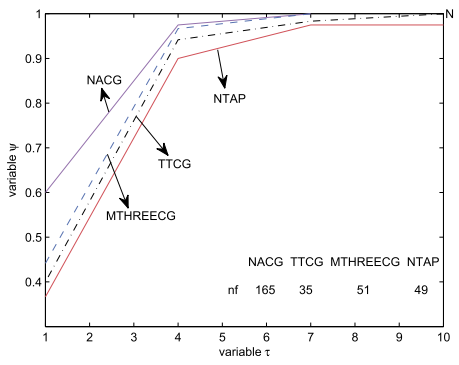
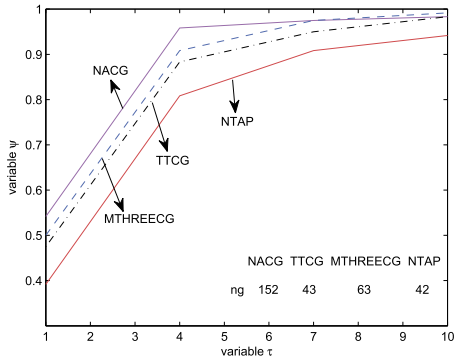


Figure 4 The number of gradient evaluations



From Fig. 1, it is obvious that the NACG exhibits the best performance subject to the number of iterations. For example, the NACG outperforms in 129 problems, the MTHREECG outperforms in 66 problems, while the other two methods outperform in 48 problems and 57 problems, respectively.

We see from Fig. 2 that the curve "MTHREECG" and "TTC" are very close, which are worse than the "NACG". The NACG occupies the first place, which solves about 57% of the 300 test problems with the least CPU time.

Figures 3 and 4 show that if the values of τ are controlled in the range of 1 to 4, the curve "NACG" is always on the top, which means that our new algorithm is competitive relative to function evaluations and gradient evaluations, respectively. Until we expand the tolerance, the performance of the MTHREECG and TTCG are almost as same as that of the NACG, while the curve "NTAP" is at the bottom all the time.

In one word, all numerical performances indicate that the efficiency and stability of the NACG is promising, even if the dimensions of the test problems exceed 5000. Moreover, we conclude that the restarted scheme is called rarely from the numerical results.

If program runs failure, or the number of iterations reaches more than 500, or precision exceeds the optimal precision in the same test problem 10^3 times or more, regarded as failed. Then we denote the number of iterations, function evaluations, gradient evaluations by 500 and CPU time by 10 seconds, respectively. In this way, the numerical results indicate that the algorithm NACG is encouraging.

5 Conclusions

Conjugate gradient methods are widely used for solving large-scale unconstrained optimization problems, due to their simplicity and low storage. We employed the idea of BFGS quasi-Newton method to improve the performance of conjugate gradient methods. Without affecting the amount of calculation and storage, the choice of the parameter in the proposed method provides more useful information. The generated search direction is close to a quasi-Newton direction and fulfills not only the sufficient descent condition, but also the Dai–Liao conjugacy condition. Furthermore, under proper conditions, we prove the global convergence of the proposed method with Wolfe line search. For a set of 300 test problems, compared with the existing effective methods, the performance profiles show that the proposed method is promising for large-scale unconstrained optimization.

It is worth emphasizing that conjugate gradient methods combining with BFGS updating technique represent an interesting computational innovation which produce efficient conjugate gradient algorithms. Our future work will be concentrated on developing some new methods to obtain superlinear convergence and extending the convergence results to general functions.

Acknowledgements
The authors are grateful to the editor and the anonymous reviewers for their valuable comments and suggestions, which have substantially improved this paper.

Funding
This work is supported by the Innovation Talent Training Program of Science and Technology of Jilin Province of China (20180519011JH), and the Science and Technology Development Project Program of Jilin Province (20190303132SF).

Availability of data and materials
Not applicable.

Competing interests
The authors declare that they have no competing interests.

Authors' contributions
The authors conceived of the study and drafted the manuscript. All authors read and approved the final version of this paper.

Author details
[1]College of Mathematics, Jilin University, Changchun, China. [2]College of Mathematics and Statistics, Beihua University, Jilin, China.

Publisher's Note

Received: 29 July 2019 Accepted: 28 October 2019 Published online: 20 November 2019

References

1. Andrei, N.: A simple three-term conjugate gradient algorithm for unconstrained optimization. J. Comput. Appl. Math. **241**, 19–29 (2013)
2. Andrei, N.: On three-term conjugate gradient algorithms for unconstrained optimization. Appl. Math. Comput. **219**, 6316–6327 (2013)
3. Andrei, N.: A new three-trem conjugate gradient algorithm for unconstrained optimization. Numer. Algorithms **68**, 305–321 (2015)
4. Babaie-Kafaki, S., Ghanbari, R.: A descent family of Dai–Liao conjugate gradient methods. Optim. Methods Softw. **29**, 583–591 (2014)
5. Babaie-Kafaki, S., Ghanbari, R.: The Dai–Liao nonlinear conjugate gradient method with optimal parameter choices. Eur. J. Oper. Res. **234**, 625–630 (2014)
6. Babaie-Kafaki, S., Ghanbari, R.: Two optimal Dai–Liao conjugate gradient methods. Optimization **64**, 2277–2287 (2014)
7. Babaie-Kafaki, S., Ghanbari, R., Mahdavi-Amiri, N.: Two new conjugate gradient methods based on modified secant equations. J. Comput. Appl. Math. **234**, 1374–1386 (2010)
8. Dai, Y.H., Liao, L.Z.: New conjugacy conditions and related nonlinear conjugate gradient methods. Appl. Math. Optim. **43**, 87–101 (2001)

9. Dai, Y.H., Yuan, Y.X.: An efficient hybrid conjugate gradient method for unconstrained optimization. Ann. Oper. Res. **103**, 33–47 (2001)
10. Dai, Z.F.: Comments on a new class of nonlinear conjugate gradient coefficients with global convergence properties. Appl. Math. Comput. **276**, 297–300 (2016)
11. Dai, Z.F., Chen, X.H., Wen, F.H.: A modified Perry's conjugate gradient method-based derivative-free method for solving large-scale nonlinear monotone equations. Appl. Math. Comput. **270**, 378–386 (2015)
12. Dai, Z.F., Chen, X.H., Wen, F.H.: Comments on "A hybrid conjugate gradient method based on a quadratic relaxation of the Dai–Yuan hybrid conjugate gradient parameter". Optimization **64**, 1173–1175 (2015)
13. Dai, Z.F., Wen, F.H.: Comments on another hybrid conjugate gradient algorithm for unconstrained optimization by Andrei. Numer. Algorithms **69**, 337–341 (2015)
14. Deng, S.H., Wan, Z.: A three-term conjugate gradient algorithm for large-scale unconstrained optimization problems. Appl. Numer. Math. **92**, 70–81 (2015)
15. Dolan, E., Moré, J.J.: Benchmarking optimization software with performance profiles. Math. Program. **91**, 201–213 (2002)
16. Flether, R., Reeves, C.M.: Function minimization by conjugate gradients. Comput. J. **7**, 149–154 (1964)
17. Ford, J.A., Narushima, Y., Yabe, H.: Multi-step nonlinear conjugate gradient methods for unconstrained minimization. Comput. Optim. Appl. **40**, 191–216 (2008)
18. Hager, W.W., Zhang, H.C.: A new conjugate gradient method with guaranteed descent and an efficient line search. SIAM J. Optim. **16**, 170–192 (2005)
19. Hestenes, M.R., Stiefel, E.L.: Methods of conjugate gradients for solving linear systems. J. Res. Natl. Bur. Stand. **49**, 409–436 (1952)
20. Huang, C.X., Yang, Z.C., Yi, T.S., Zou, X.F.: On the basins of attraction for a class of delay differential equations with non-monotone bistable nonlinearities. J. Differ. Equ. **256**, 2101–2114 (2014)
21. Kou, C.X.: An improved nonlinear conjugate gradient method with an optimal property. Sci. China Math. **57**, 635–648 (2014)
22. Li, D.H., Fukushima, M.: A modified BFGS method and its global convergence in nonconvex minimization. J. Comput. Appl. Math. **129**, 15–35 (2001)
23. Liu, C.Y., Gong, Z.H., Teo, K.L., Sun, J., Caccetta, L.: Robust multi-objective optimal switching control arising in 1, 3-propanediol microbial fed-batch process. Nonlinear Anal. Hybrid Syst. **25**, 1–20 (2017)
24. Liu, S., Chen, Y.P., Huang, Y.Q., Zhou, J.: An efficient two grid method for miscible displacement problem approximated by mixed finite element methods. Comput. Math. Appl. **77**, 752–764 (2019)
25. Livieris, I.E., Pintelas, P.: A descent Dai–Liao conjugate gradient method based on a modified secant equation and its global convergence. ISRN Comput. Math. **2012**, Article ID 435295 (2012)
26. Narushima, Y., Yabe, H.: Conjugate gradient methods based on secant conditions that generate descent search directions for unconstrained optimization. J. Comput. Appl. Math. **236**, 4303–4317 (2012)
27. Perry, A.: Technical note—a modified conjugate gradient algorithm. Oper. Res. **26**, 1073–1078 (1978)
28. Polak, E., Ribiére, G.: Note sur la convergence des méthodes de directions conjuguées. Rev. Fr. Inform. Rech. Oper., 3e Année **16**, 35–43 (1969)
29. Polyak, B.T.: The conjugate gradient method in extreme problems. USSR Comput. Math. Math. Phys. **9**, 94–112 (1969)
30. Sugiki, K., Narushima, Y., Yabe, H.: Globally convergent three-term conjugate gradient methods that use secant conditions and generate descent search directions for unconstrained optimization. J. Optim. Theory Appl. **153**, 733–757 (2012)
31. Wang, J.F., Chen, X.Y., Huang, L.H.: The number and stability of limit cycles for planar piecewise linear systems of node-saddle type. J. Math. Anal. Appl. **469**, 405–427 (2019)
32. Wang, J.F., Huang, C.X., Huang, L.H.: Discontinuity-induced limit cycles in a general planar piecewise linear system of saddle-focus type. Nonlinear Anal. Hybrid Syst. **22**, 162–178 (2019)
33. Wolfe, P.: Convergence conditions for ascent methods. SIAM Rev. **11**, 226–235 (1969)
34. Wolfe, P.: Convergence conditions for ascent methods, II: some corrections. SIAM Rev. **13**, 185–188 (1971)
35. Yabe, H., Takano, M.: Global convergence properties of nonlinear conjugate gradient methods with modified secant condition. Comput. Optim. Appl. **28**, 203–225 (2004)
36. Yang, Y.T., Chen, Y.T., Lu, Y.L.: A subspace conjugate gradient algorithm for large-scale unconstrained optimization. Numer. Algorithms **76**, 813–828 (2017)
37. Yao, S.W., Ning, L.S.: An adaptive three-term conjugate gradient method based on self-scaling memoryless BFGS matrix. J. Comput. Appl. Math. **322**, 72–85 (2018)
38. Yuan, J.L., Zhang, Y.D., Ye, J.X., Xie, J., Teo, K.L., Zhu, X., Feng, E.M., Yin, H.C., Xiu, Z.L.: Robust parameter identification using parallel global optimization for a batch nonlinear enzyme-catalytic time-delayed process presenting metabolic discontinuities. Appl. Math. Model. **46**, 554–571 (2017)
39. Zhang, L., Jian, S.Y.: Further studies on the Wei–Yao–Liu nonlinear conjugate gradient method. Appl. Math. Comput. **219**, 7616–7621 (2013)
40. Zhou, W.J.: A short note on the global convergence of the unmodified PRP method. Optim. Lett. **7**, 1367–1372 (2013)
41. Zhou, W.J.: On the convergence of the modified Levenberg–Marquardt method with a nonmonotone second order Armijo type line search. J. Comput. Appl. Math. **239**, 152–161 (2013)
42. Zhou, W.J., Chen, X.L.: On the convergence of a modified regularized Newton method for convex optimization with singular solutions. J. Comput. Appl. Math. **239**, 179–188 (2013)
43. Zhou, W.J., Shen, D.M.: An inexact PRP conjugate gradient method for symmetric nonlinear equations. Numer. Funct. Anal. Optim. **35**, 370–388 (2014)
44. Zhou, W.J., Zhang, L.: A nonlinear conjugate gradient method based on the MBFGS secant condition. Optim. Methods Softw. **21**, 707–714 (2006)
45. Zoutendijk, G.: Nonlinear programming, computational method. In: Abadie, J. (ed.) Integer and Nonlinear Programming, pp. 37–86. North-Holland, Amsterdam (1970)