Yokohama Publishers
ISSN 1349-8169 ONLINE JOURNAL

Pacific Journal of Optimization
Vol. 15, No. 2, pp. 237 – 248, 2019

A NEW DAI-LIAO TYPE OF CONJUGATE GRADIENT ALGORITHM FOR UNCONSTRAINED OPTIMIZATION PROBLEMS*

Songhai Deng, Jing Lv and Zhong Wan†

Abstract: In this paper, a new Dai-Liao type of three-term conjugate gradient algorithm is developed for solving nonconvex unconstrained optimization problems. The search direction consists of three terms, which aim to gather more useful information of the current iterate point such that the direction has better convergence performance for the algorithm. Different from the existing methods, global convergence is established without assumption of uniform convexity under a modified Armijo-type line search. Numerical experiments are employed to show efficiency of the algorithm in solving large-scale benchmark test problems, especially in comparison with the state-of-the-art ones in the literature.

Key words: *unconstrained optimization, three-term conjugate gradient method, global convergence, Armijo-type line search, algorithm*

Mathematics Subject Classification: *90C25, 90C30*

1 Introduction

Due to lower computational cost and lower memory requirements, conjugate gradient methods are widely used for solving large-scale unconstrained and constrained optimization problems [14, 15, 18, 22]. In these methods, the way to choose a better conjugate parameter plays a fundamental role in determining a high-quality search direction. In [12], eight choices of conjugate parameters were presented.

Recently, three-term conjugate gradient algorithms are attracting the interest of many scholars. For a part of results in this connection, one can see [1, 2, 7, 8, 19, 25, 26] and the references therein. As an improvement of conjugate gradient method, the three-term conjugate gradient method is often a linear combination of negative gradient, iteration direction at the last direction and a correction vector such as the difference of gradients [1, 2].

In this paper, we intend to study a new three-term conjugate gradient method, where the conjugate parameter looks like the Dai-Liao-Type in [6] or the Perry-Type in [21]. Furthermore, we attempt to establish global convergence of this method combined with a modified Armijo-type line search, rather than the Wolfe-type line searches as used in the

*This work is supported by National Natural Science Foundation of China (Grant No. 71671190, 71631008) and Opening Foundation from State Key Laboratory of Developmental Biology of Freshwater Fish, Hunan Normal University..

†Corresponding author.

literature. Numerical experiments will be employed to show its efficiency in solving large-scale problems.

The rest of this paper is organized as follows. In next section, a new three-term Dai-Liao-Type conjugate gradient method is proposed. Then, a new algorithm is developed and its global convergence is also established in Section 3. Section 4 is devoted to numerical experiments. Some conclusions are drawn in the last section.

2 Design of a New Three-Term Conjugate Gradient Method

Consider the following unconstrained optimization problem:

$$\min f(x), \quad x \in \mathbb{R}^n, \tag{2.1}$$

where $f : \mathbb{R}^n \to \mathbb{R}$ is continuously differentiable such that its gradient is available. Let $g : \mathbb{R}^n \to \mathbb{R}^n$ denote the gradient function of f, and let g_k denote the value of g at x_k.

Let $x_0 \in \mathbb{R}^n$ be an initial point. A sequence of approximate solutions of (2.1), $\{ x_k \}$, is often generated by

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

where $k \geq 0$, α_k is called a step size obtained by some line search rule and d_k is a search direction [17]. In the classical conjugate gradient methods, d_k is given by

$$d_k = \begin{cases} -g_k, & \text{if } k = 0, \\ -g_k + \beta_k d_{k-1}, & \text{if } k > 0. \end{cases} \tag{2.2}$$

In (2.2), β_k is called the conjugate parameter. With a different choice of β_k, the obtained method has distinct numerical performance. In [12], the following eight choices are presented:

$$\begin{aligned}
&\beta_k^{HS} = \frac{g_{k+1}^T y_k}{d_k^T y_k}, && \beta_k^{FR} = \frac{\|g_{k+1}\|^2}{\|g_k\|^2}, \\
&\beta_k^{D} = \frac{g_{k+1}^T \nabla^2 f_k d_k}{d_k^T \nabla^2 f_k d_k}, && \beta_k^{PRP} = \frac{g_{k+1}^T y_k}{\|g_k\|^2}, \\
&\beta_k^{CD} = \frac{\|g_{k+1}\|^2}{-d_k^T g_k}, && \beta_k^{LS} = \frac{g_{k+1}^T y_k}{-d_k^T g_k}, \\
&\beta_k^{DY} = \frac{\|g_{k+1}\|^2}{d_k^T y_k}, && \beta_k^{HZ} = \frac{1}{d_k^T y_k}\left(y_k - 2 d_k \frac{\|y_k\|^2}{d_k^T y_k} \right)^T g_{k+1},
\end{aligned}$$

where $y_k = g_{k+1} - g_k$, $\| \cdot \|$ is the Euclidean norm, defined by $\|a\| = \sqrt{\sum_{i=1}^n a_i^2}$ for an n-dimensional vector $a = (a_1, \ldots, a_n)^T$, and $\nabla^2 f_k$ is the Hessian matrix of the objective function at x_k. Based on the above eight forms, there have been a lot of weighted β_k constructed by using different denominators and numerators. It has been proved [13] that β_k^{HZ} can ensure that the obtained search direction d_k is sufficiently descent, and if $d_k^T y_k \neq 0$, then $g_k^T d_k \leq -\frac{7}{8}\|g_k\|^2$ holds. On this basis, an algorithm, called the CG_DESCENT, is developed with a specific line search strategy. It has been reported that the numerical performance of the CG_DESCENT algorithm is impressive, particularly for solving large-scale problems.

For development of new algorithms, Dai and Liao [6] proposed a new conjugacy condition as follows:

$$d_{k+1}^T y_k = -t_k g_{k+1}^T s_k, \tag{2.3}$$

where $s_k = x_{k+1} - x_k (= \alpha_k d_k)$ and $t_k > 0$. Then, a conjugate gradient algorithm (DL) was developed where the conjugate parameter is

$$\beta_k^{DL} = \frac{g_k^T (y_{k-1} - t s_{k-1})}{d_{k-1}^T y_{k-1}}.$$

Given different values of t, different algorithms can be developed. Clearly, if $t = \frac{2\|y_k\|^2}{s_k^T y_k}$, then $\beta_k^{DL} = \beta_k^{HZ}$.

Recently, Deng and Wan [9] constructed a new form of β_k as follows:

$$\beta_k = \frac{g_k^T (y_{k-1} - s_{k-1})}{d_{k-1}^T \left(I - \frac{g_k g_k^T}{\|g_k\|^2} \right) y_{k-1}}. \tag{2.4}$$

Denote

$$\overline{y_{k-1}} = \left(I - \frac{g_k g_k^T}{\|g_k\|^2} \right) y_{k-1}. \tag{2.5}$$

Then,

$$\beta_k = \frac{g_k^T (y_{k-1} - s_{k-1})}{d_{k-1}^T \overline{y_{k-1}}} \tag{2.6}$$

is very similar to

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

For this reason, we call β_k in (2.6) a conjugate parameter of the Dai-Liao-Type. It has been shown [9] that this β_k can improve efficiency of the algorithm as y_{k-1} is replaced by $\overline{y_{k-1}}$.

On the other hand, in order to further improve the efficiency of the classical conjugate gradient method, a type of three-term conjugate gradient methods have been presented and widely studied.

The first general three-term conjugate gradient method was proposed in [4], which determines the search direction as follows:

$$d_{k+1} = -g_{k+1} + \beta_k d_k + \gamma_k d_t, \tag{2.7}$$

where $\beta_k = \beta_k^{HS}$ (or β_k^{FR}, β_k^{DY} etc.), d_t ($t \leq k-1$) is a restart direction, and

$$\gamma_k = \begin{cases} 0, & t = k-1, \\ \dfrac{g_{k+1}^T y_t}{d_t^T y_t}, & t < k-1. \end{cases} \tag{2.8}$$

Then, in [20], Nazareth developed another three-term conjugate gradient algorithm, where the search direction is given by

$$d_{k+1} = -y_k + \frac{y_k^T y_k}{y_k^T d_k} d_k + \frac{y_{k-1}^T y_k}{y_{k-1}^T d_{k-1}} d_{k-1} \tag{2.9}$$

with $d_{-1} = d_0 = 0$. It has been proved that if f is a convex quadratic function, then for any stepsize α_k, the search directions generated by (2.9) are conjugate with respect to the coefficient matrix of quadratic term. In [26], a descent modified Polak-Ribiére-Polyak (PRP) conjugate gradient algorithm was developed, where the search direction was obtained by the following three-term formula:

$$d_{k+1} = -g_{k+1} + \frac{g_{k+1}^T y_k}{g_k^T g_k} d_k - \frac{g_{k+1}^T d_k}{g_k^T g_k} y_k.$$

A remarkable property of the above methods is that the constructed directions are sufficiently descent, i.e., it satisfies that $g_k^T d_k = -\|g_k\|^2$ for any $k \geq 0$.

Andrei in [1,2] investigated the following two types of descent three-term gradient methods:

$$d_{k+1} = -g_{k+1} - \left(\left(1 + \frac{\|y_k\|^2}{y_k^T s_k} \right) \frac{s_k^T g_{k+1}}{y_k^T s_k} - \frac{y_k^T g_{k+1}}{y_k^T s_k} \right) s_k - \frac{s_k^T g_{k+1}}{y_k^T s_k} y_k, \tag{2.10}$$

$$d_{k+1} = -g_{k+1} - \left(\left(1 + 2\frac{\|y_k\|^2}{y_k^T s_k} \right) \frac{s_k^T g_{k+1}}{y_k^T s_k} - \frac{y_k^T g_{k+1}}{y_k^T s_k} \right) s_k - \frac{s_k^T g_{k+1}}{y_k^T s_k} y_k. \tag{2.11}$$

It has been shown that the two search directions in (2.10) and (2.11) satisfy the Dai-Liao's conjugacy condition (2.3).

Motivated by the ideas in [1,9], we choose a search direction by

$$d_{k+1} = -g_{k+1} + \frac{g_{k+1}^T (y_k - s_k)}{d_k^T \overline{y_k}} d_k + \frac{g_{k+1}^T d_k}{d_k^T \overline{y_k}} (s_k - y_k), \tag{2.12}$$

where $\overline{y_k}$ is defined as in (2.5), and I is an n-th order unit matrix.

As pointed in [9], $d_k^T \overline{y_k}$ is not always greater than 0. To overcome this disadvantage, as done in [9], we can modify

$$\beta_k = \begin{cases} \dfrac{g_{k+1}^T (y_k - s_k)}{d_k^T \overline{y_k}}, & \text{if} \quad d_k^T \overline{y_k} > \eta \|g_{k+1}\|^2; \\ \dfrac{g_{k+1}^T y_k}{\|g_{k+1}\|^2} = \beta_k^{PRP}, & \text{otherwise.} \end{cases} \tag{2.13}$$

In this paper, replacing the piecewise format (2.13), a weighted denominator is used. It says that

$$\beta_k = \frac{g_{k+1}^T (y_k - s_k)}{|d_k^T \overline{y_k}| + \mu \|g_{k+1}\|^2}, \tag{2.14}$$

and

$$\theta_k = \frac{g_{k+1}^T d_k}{|d_k^T \overline{y_k}| + \mu \|g_{k+1}\|^2}, \tag{2.15}$$

where $\mu > 0$ is a constant. Consequently, the proposed new three-term conjugate gradient method in this paper determines a search direction by

$$d_{k+1} = -g_{k+1} + \beta_k d_k + \theta_k (s_k - y_k), \tag{2.16}$$

where β_k and θ_k are defined by (2.14) and (2.15), respectively.

3 Development of Algorithm and its Convergence

In this section, we shall develop a new algorithm and analyze its convergence.

$\boxed{3.1}$ Development of algorithm

Due to requirement of establishing global convergence, many conjugate gradient algorithms need the Wolfe line search to choose a step size, rather than the Armijo line search with more lower computational cost (see [10, 16] and the references therein). Specifically, for the Armijo step size α_k, only the following inequality is required to be satisfied:

$$f(x_{k+1}) = f(x_k + \alpha_k d_k) \leq f(x_k) + \delta \alpha_k g(x_k)^T d_k. \tag{3.1}$$

If the Wolfe step size is adopted, then one needs to find an α_k such that the following two inequalities are simultaneously satisfied:

$$\begin{cases} f(x_{k+1}) = f(x_k + \alpha_k d_k) \leq f(x_k) + \delta \alpha_k g(x_k)^T d_k \\ g(x_{k+1})^T d_k \geq \sigma g(x_k)^T d_k. \end{cases} \tag{3.2}$$

If (3.2) is replaced by

$$\begin{cases} f(x_{k+1}) \leq f(x_k) + \delta g(x_k)^T d_k \\ |g(x_{k+1})^T d_k| \leq \sigma |g(x_k)^T d_k|, \end{cases} \tag{3.3}$$

then the step size satisfies the strong Wolfe conditions.

The three-term conjugate gradient methods (2.10) and (2.11) proposed in [1, 2, 10, 16] were proved to be globally convergent in the case that the step size satisfies the strong Wolfe conditions for the strong convex or uniquely convex optimization problems. As the step size is obtained by the Armijo line search or the optimization is nonconvex, establishment of global convergence is often difficult for the conjugate gradient algorithms [23, 24]. As one of main contributions in this paper, we attempt to prove the global convergence of our algorithm (see Algorithm 3.1) under the following modified Armijo-type line search:

$$f(x_{k+1}) < f(x_k) + \delta_1 \alpha_k g_k^T d_k - \delta_2 \alpha_k^2 \|d_k\|^2 \tag{3.4}$$

With the above preparation, we are in a position to present an overall framework of our algorithm.

Algorithm 3.1 (New Dai-Liao Type of Three-term Conjugate Gradient Algorithm (DLTTCG))**.**

Step 1. Select a starting point $x_0 \in \mathrm{dom} f$ and compute $f_0 = f(x_0)$ and $g_0 = \nabla g(x_0)$, $d_0 = -g_0$. Set $k := 0$.

Step 2. If $\|g_k\|_\infty < \epsilon$, then the algorithm stops. Otherwise, go to Step 3.

Step 3. Determine a step size α_k by the line search (3.4).

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

Step 5. Compute β_k and θ_k as defined by (2.14) and (2.15), respectively.

Step 6. Determine a new search direction by (2.16).

Step 7. Set $k := k + 1$. Return to Step 2.

In Algorithm 3.1, $\|\cdot\|_\infty$ denotes the infinity norm of a vector, defined by

$$\|x\|_\infty := \max_{1 \leq k \leq n} |x_k|.$$

$\boxed{3.2}$ Global convergence

In this section, we are going to study the global convergence of Algorithm 3.1.

We first state the following mild assumptions, which are required to prove the main results in this paper.

Assumption 3.2. The level set $\Omega = \{x \in \mathbb{R}^n \mid f(x) \leq f(x_0)\}$ is bounded.

Assumption 3.3. In some neighborhood N of Ω, f is continuously differentiable and its gradient is Lipschitz continuous, namely, there exists a constant $L > 0$ such that

$$\|g(x) - g(y)\| \leq L\|x - y\|, \quad \forall x, y \in N. \tag{3.5}$$

Since the sequence $\{\, f(x_k)\,\}$ is decreasing, the sequence $\{x_k\}$ generated by Algorithm 3.1 is clearly contained in a bounded region by Assumption 3.2. Therefore, there exists a convergent subsequence of $\{x_k\}$. Without loss of generality, we suppose that $\{x_k\}$ is convergent. On the other hand, from Assumptions 3.2 and 3.3, it is easy to see that there is a constant $\gamma_1 > 0$ such that $\|g(x)\| \leq \gamma_1$, $\forall x \in \Omega$. Thus, the sequence $\{\, g_k\,\}$ is bounded.

Proposition 3.4. *Let $f : \mathbb{R}^n \to \mathbb{R}$ be a continuously differentiable function. Suppose that d is a descent direction of f at x. Then, there exists a nonnegative integer number j_0 such that*

$$f(x + \alpha d) \leq f(x) + \delta_1 \alpha g^T d - \delta_2 \alpha^2 \|d\|^2, \tag{3.6}$$

where $\alpha = \rho^{j_0}$, g is the gradient of f at x, all of $\delta_1 > 0$, $\delta_2 > 0$ and $\rho \in (0,1)$ are given constant scalars.

Proof. The proof can be completed similar to [9].

Lemma 3.5. *Let d_{k+1} be given by* (2.16)*, where β_k be given by* (2.14) *and θ_k be given by* (2.15)*. Then, the following equality*

$$g_{k+1}^T d_{k+1} = -\|g_{k+1}\|^2 \tag{3.7}$$

holds for any $k \geq 0$.

Proof. By the formulas (2.14), (2.15) and (2.16), we have

$$d_{k+1} = -g_{k+1} + \beta_k d_k + \theta_k (s_k - y_k),$$

where

$$\beta_k = \frac{g_{k+1}^T (y_k - s_k)}{|d_k^T \overline{y_k}| + \mu \|g_{k+1}\|^2},$$

and

$$\theta_k = \frac{g_{k+1}^T d_k}{|d_k^T \overline{y_k}| + \mu \|g_{k+1}\|^2}.$$

It is clear that

$$\begin{aligned} g_{k+1}^T d_{k+1} &= -\|g_{k+1}\|^2 + \frac{g_{k+1}^T (y_k - s_k)}{|d_k^T \overline{y_k}| + \mu \|g_{k+1}\|^2} g_{k+1}^T d_k + \frac{g_{k+1}^T d_k}{|d_k^T \overline{y_k}| + \mu \|g_{k+1}\|^2} g_{k+1}^T (s_k - y_k) \\ &= -\|g_{k+1}\|^2. \end{aligned}$$

Furthermore, from

$$\|g_{k+1}\|^2 = |g_{k+1}^T d_{k+1}| \leq \|g_{k+1}\| \|d_{k+1}\|,$$

it follows that

$$\|g_{k+1}\| \leq \|d_{k+1}\|.$$

Lemma 3.6 (see [9]). *Let $\{ \alpha_k \}$ and $\{ d_k \}$ be the two sequences of step lengths and search directions generated by Algorithm* 3.1*, respectively. Then,*

$$\lim_{k\to\infty} \alpha_k^2 \|d_k\|^2 = 0.$$

Lemma 3.7. *Let the search direction d_{k+1} be given by* (2.16)*, where β_k is computed as in* (2.14)*, and θ_k is computed as in* (2.15)*. Assume that $g_k \geq \epsilon$ for all $k > 0$. Then, $\|d_{k+1}\| \leq M$.*

Proof. Without loss of generality, take $0 < \epsilon < 1$. By Lemma 3.6, we have

$$\lim_{k\to\infty} \alpha_k \|d_k\| = 0.$$

It implies that there exists an $N > 0$ large enough such that as $k > N$, it holds that

$$\alpha_k \|d_k\| < \frac{\mu\epsilon^3}{2\gamma_1(L+1)}.$$

Thus,

$$\begin{aligned}
\|d_{k+1}\| &\leq \|g_{k+1}\| + \frac{\|g_{k+1}\|(\|y_k\| + \|s_k\|)\|}{\mu\|g_{k+1}\|^2}\|d_k\| + \frac{\|g_{k+1}\|\|d_k\|}{\mu\|g_{k+1}\|^2}(\|y_k\| + \|s_k\|) \\
&\leq \gamma_1 + \frac{2\gamma_1(L+1)\alpha_k\|d_k\|}{\mu\epsilon^2}\|d_k\| \\
&< \gamma_1 + \epsilon\|d_k\| \\
&< \gamma_1 + \epsilon(\gamma_1 + \epsilon\|d_{k-1}\|) \\
&< \gamma_1 + \gamma_1\epsilon + \epsilon^2\|d_{k-1}\| \\
&\vdots \\
&< \gamma_1(1 + \epsilon + \epsilon^2 + \ldots + \epsilon^{k-N}) + \epsilon^{k-N+1}\|d_N\| \\
&< \frac{\gamma_1}{1-\epsilon} + \|d_N\| = M_1.
\end{aligned} \tag{3.8}$$

Set $M = \max\{\|d_1\|, \ldots, \|d_{N-1}\|, M_1\}$. Then, $d_k < M$ for all $k > 0$. □

The following result, often being called the Zoutendijk condition, is often used to prove global convergence of many conjugate gradient methods in the literature. It was first given by Zoutendijk [27]. For Algorithm 3.1, we can prove that it also holds.

Lemma 3.8. *Under Assumptions 3.2 and 3.3, it holds that*

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

Proof. From the line search rule (3.4) and Assumption 3.2, there exists a constant M such that

$$\sum_{k=0}^{n-1} \left(-\delta_1\alpha_k g_k^T d_k + \delta_2\alpha_k^2\|d_k\|^2\right) \leq \sum_{k=0}^{n-1} (f(x_k) - f(x_{k+1})) = f(x_0) - f(x_n) < 2M.$$

With Assumption 3.3, there exists a constant $m > 0$ such that the inequality

$$\alpha_k \geq m\frac{\|g_k\|^2}{\|d_k\|^2}$$

holds for all k sufficiently large (The proof can be completed as in [9]). Then, from Lemma 3.5 and the last inequality, we have

$$\begin{aligned} 2M &> \sum_{k=0}^{n-1} \left(-\delta_1 \alpha_k g_k^T d_k + \delta_2 \alpha_k^2 \|d_k\|^2\right) \\ &= \sum_{k=0}^{n-1} \left(\delta_1 \alpha_k \|g_k\|^2 + \delta_2 \alpha_k^2 \|d_k\|^2\right) \\ &\geq \sum_{k=0}^{n-1} \left(\delta_1 m \frac{\|g_k\|^2}{\|d_k\|^2} \|g_k\|^2 + \delta_2 m^2 \frac{\|g_k\|^4}{\|d_k\|^4} \|d_k\|^2\right) \\ &= \sum_{k=0}^{n-1} (\delta_1 + \delta_2 m) \frac{\|g_k\|^4}{\|d_k\|^2} m. \end{aligned}$$

Thus, the desired result has been proved.

With the above preparation, we are in a position to state the main result in this paper.

Theorem 3.9. *Suppose that f in Problem (2.1) is continuous differentiable. Let $\{ g_k \}$ be the gradient sequence generated by Algorithm* 3.1. *Under Assumptions 3.2 and 3.3, the following result holds:*

$$\liminf_{k \to \infty} \|g_k\| = 0. \tag{3.9}$$

Proof. Suppose that (3.9) does not hold true, then there exists a positive $\epsilon > 0$ such that for all k, $\|g_k\| \geq \epsilon$. It follows from Lemma 3.7 that

$$\frac{\|g_k\|^4}{\|d_k\|^2} > \frac{\epsilon^4}{M^2}.$$

Therefore, the series $\sum_{k=0}^{\infty} \frac{\|g_k\|^4}{\|d_k\|^2}$ diverges, which contradicts the result of Lemma 3.8. We have completed the proof. □

4 Numerical Experiments

In this section, we shall report the numerical performance of Algorithm 3.1.

We test Algorithm 3.1 (DLTTCG) by using it to solve the 75 benchmark test problems from [3], some of them are from CUTEr [5], and the dimension of these problems changes from 1000 to 10000.

We compare its numerical performance with the spectral conjugate gradient method [9] (ISCG), improved three-term conjugate gradient method [7] (ITTCG), the three-term conjugate gradient algorithm TTCG in [2], which has been reported to be very efficient for solution of nonconvex unconstrained optimization problems. Among these algorithms, either the search direction or the line search strategy is different from each other.

The code of the computer procedure is written in Fortran 77, and is implemented on PC with 2.2 GHz CPU processor, 2 GB RAM memory.

The parameters in Algorithm 3.1 and those in ISCG are specified by

$$\epsilon = 10^{-6}, \quad \rho = 0.3, \quad \delta_1 = 0.4, \quad \delta_2 = 0.001, \quad \mu = 0.01.$$

We report a part of numerical results in Table 1. In Table 1, we use the following notations:

DIM: the number of the decision variables;
NI: the number of iterations;
NF: the number of function evaluations;
NG: the number of gradient evaluations;

Table 1: Comparison of efficiency with other similar algorithms on several functions

Function name		DIM	NI	NF	NG	CT (s)	$f(x^*)$	$\|g\|_\infty$
Extended Trigonometric ET1	DLTTCG	1000	23	69	15	1	0.206544E-10	0.526199E-05
	TTCG	3000	27	83	19	4	0.497671E-12	0.114275E-05
	ISCG	3000	39	332	40	91	0.234521E-11	0.981482E-06
	ITTCG	3000	26	83	21	7	0.561228E-11	0.273425E-05
Extended Beale(CUTE)	DLTTCG	1000	11	33	10	0	0.110914E-10	0.258598E-05
	TTCG	1000	12	34	9	0	0.503148E-11	0.221926E-04
	ISCG	1000	37	257	38	3	0.825614E-09	0.922361E-06
	ITTCG	1000	13	38	11	0	0.304568E-14	0.428524E-07
Extended Penalty	DLTTCG	4000	10	33	10	1	0.370407E+04	0.132389E-05
	TTCG	4000	10	33	10	2	0.370407E+04	0.130330E-05
	ISCG	3000	63	640	64	18	0.275597E+04	0.977183E-06
	ITTCG	4000	12	67	12	3	0.370407E+04	0.268024E-06
Perturbed Quadratic	DLTTCG	1000	175	393	43	4	0.168679E-12	0.758708E-05
	TTCG	1000	175	393	42	3	0.168679E-12	0.758708E-05
	ISCG	1000	308	2079	309	25	0.199440E-12	0.805935E-06
	ITTCG	1000	178	400	43	3	0.939630E-13	0.478450E-05
Raydan 1	DLTTCG	5000	456	956	43	65	0.125025E+07	0.902263E-05
	TTCG	5000	460	962	41	43	0.125025E+07	0.898480E-05
	ISCG	5000	352	523	53	33	0.200100E+06	0.398174E-06
	ITTCG	5000	456	956	43	80	0.125025E+07	0.902262E-05
Raydan 2	DLTTCG	10000	3	10	3	0	0.100000E+05	0.102475E-06
	TTCG	10000	3	10	3	2	0.100000E+05	0.102474E-06
	ISCG	10000	11	59	12	25	0.100000E+05	0.279742E-06
	ITTCG	10000	3	10	3	1	0.100000E+05	0.102474E-06
Generalized Tridiagonal 1	DLTTCG	1000	21	65	18	0	0.997210E+03	0.172830E-05
	TTCG	1000	21	65	18	2	0.997210E+03	0.166437E-05
	ISCG	1000	40	292	41	5	0.997210E+03	0.968906E-06
	ITTCG	1000	21	65	18	3	0.997210E+03	0.168276E-05
Extended Tridiagonal 1	DLTTCG	6000	7	25	7	1	0.211515E-06	0.817618E-05
	TTCG	6000	7	25	7	2	0.208347E-06	0.807885E-05
	ISCG	6000	20	137	21	8	0.742657E-07	0.447987E-06
	ITTCG	6000	7	25	7	2	0.208347E-06	0.807876E-05
Extended Three Expo Terms	DLTTCG	1000	6	16	3	0	0.127963E+04	0.416950E-06
	TTCG	1000	6	17	4	2	0.127963E+04	0.287946E-06
	ISCG	1000	29	184	30	24	0.127963E+04	0.394938E-06
	ITTCG	1000	6	17	4	0	0.127963E+04	0.120028E-06
Extended Powell	DLTTCG	1000	148	420	106	6	0.998722E+03	0.108036E-05
	TTCG	1000	200	575	159	8	0.998722E+03	0.160087E-05
	ISCG	1000	228	1343	229	91	0.998722E+03	0.931805E-06
	ITTCG	1000	160	465	124	10	0.998722E+03	0.142503E-05
Extended PSC1	DLTTCG	3000	29	82	23	1	0.228334E-06	0.829745E-05
	TTCG	3000	47	132	37	3	0.452658E-06	0.169631E-04
	ISCG	3000	165	1208	166	18	0.116912E-07	0.449221E-06
	ITTCG	3000	29	82	23	3	0.231706E-06	0.904152E-05
Extended Block-Diagonal BD1	DLTTCG	7000	15	48	14	3	0.206364E-09	0.199160E-04
	TTCG	7000	15	47	13	1	0.337463E-09	0.254638E-04
	ISCG	7000	34	676	35	176	0.695153E-09	0.568374E-06
	ITTCG	7000	14	43	11	3	0.336856E-10	0.965515E-05
Full Hessian FH1	DLTTCG	4000	6	24	5	0	0.399573E+03	0.119886E-08
	TTCG	4000	7	22	6	2	0.399573E+03	0.139806E-06
	ISCG	4000	25	51	16	42	0.365311E-10	0.938157E-06
	ITTCG	4000	7	22	6	1	0.399573E+03	0.139806E-06
Extended Cliff	DLTTCG	1000	94	228	39	1	0.322069E-10	0.665089E-05
	TTCG	1000	95	231	40	3	0.321849E-10	0.665840E-05
	ISCG	1000	74	589	75	8	0.311101E-10	0.936330E-06
	ITTCG	1000	95	228	37	2	0.321849E-10	0.664675E-05
Quadratic Diagonal Perturbed	DLTTCG	1000	23	70	22	2	0.216079E-11	0.117293E-04
	TTCG	1000	32	93	28	0	0.229541E-10	0.981426E-05
	ISCG	1000	187	1409	188	11	0.434022E-11	0.436977E-06
	ITTCG	1000	28	82	25	2	0.163897E-12	0.175115E-04
NONDQUAR	DLTTCG	9000	9	33	9	2	0.359900E+05	0.136621E-09
	TTCG	9000	10	83	10	3	0.359900E+05	0.732281E-07
	ISCG	1000	36	307	37	3	0.399000E+04	0.805117E-06
	ITTCT	9000	10	83	10	6	0.359900E+05	0.732282E-07
Tridiagonal White &(c=4) Holst	DLTTCG	7000	35	118	29	10	0.277561E-16	0.210357E-06
	TTCG	7000	36	123	32	10	0.548907E-17	0.937149E-07
	ISCG	7000	61	519	62	25	0.741549E-11	0.990192E-06
	ITTCG	7000	37	127	34	15	0.172696E-20	0.165586E-08
Diagonal Double Borded	DLTTCG	1000	336	716	43	6	-0.100012E+01	0.434333E-05
	TTCG	1000	335	716	45	4	-0.100012E+01	0.463235E-05
	ISCG	1000	336	3171	337	18	-0.100012E+01	0.985584E-06
	ITTCG	1000	335	716	45	6	-0.100012E+01	0.464332E-05
TRIDIA (CUTE)	DLTTCG	5000	2	6	1	0	0.396588E+05	0.137555E-09
	TTCG	5000	2	6	1	0	0.396588E+05	0.137555E-09
	ISCG	1000	15	61	6	5	0.793176E+04	0.393404E-06
	ITTCG	5000	2	6	1	0	0.396588E+05	0.137555E-09
ARWHEAD (CUTE)	DLTTCG	4000	30	84	17	1	0.155852E+04	0.173839E-05
	TTCG	4000	30	121	18	3	0.155852E+04	0.174265E-05
	ISCG	1000	35	203	36	1	0.389339E+03	0.883496E-06
	ITTCG	4000	30	159	20	5	0.155853E+04	0.173891E-05
NONDIA (CUTE)	DLTTCG	2000	90	393	73	6	0.798942E+04	0.107606E-05
	TTCG	2000	96	867	87	10	0.798942E+04	0.137126E-05
	ISCG	1000	376	235	377	89	0.327853E-12	0.993499E-06
	ITTCG	2000	86	636	79	14	0.798943E+04	0.240097E-05
BDQRTIC (CUTE)	DLTTCG	1000	6	19	5	0	0.205311E-18	0.287143E-06
	TTCG	1000	7	22	6	0	0.921531E-22	0.190671E-10
	ISCG	1000	52	36	53	4	0.355614E-13	0.429237E-06
	ITTCG	1000	7	22	6	0	0.921532E-22	0.190672E-10

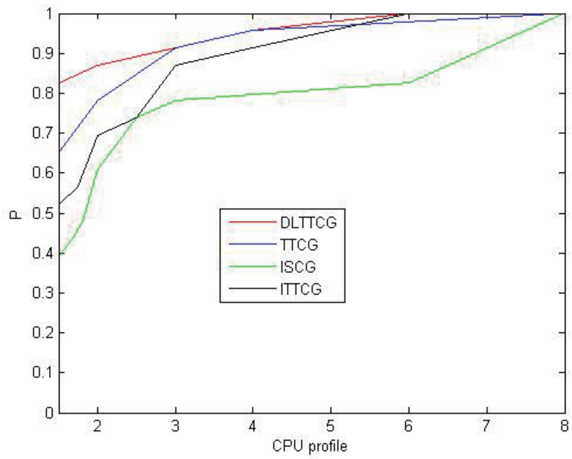

Figure 1: Numerical performance of algorithm: CPU profile

CT: the consumed CPU time(s) in the PC (in seconds);

From the results in Table 1 and the performance profile of iteration in Figure 1, it is easy to see that all the DLTTCG, TTCG,ITTCG and ISCG achieve the specified tolerance ($\|g_k\| < \epsilon$). The average numerical efficiency of the DLTTCG is better than the other three algorithms.

5 Conclusions

In this paper, we have proposed a new Dai-Liao type three-term conjugate gradient method to solve nonlinear unconstrained optimization problems, where the search direction is always sufficiently descent and the function is nonconvex.

Compared with the similar methods available in the literature, the theory of global convergence has been established without assumption of strong convexity or uniform convexity, and it is done under the modified Armijo line search, rather than the Wolfe line search.

Numerical experiments have shown the efficiency of the developed algorithm in this paper for solving large-scale benchmark test problems. The results indicates that our algorithm is promising.

Acknowledgements

The authors would like to express their thanks to the anonymous referees for their constructive comments on the paper, which have greatly improved its presentation.

References

[1] N. Andrei, A simple three-term conjugate gradient algorithm for unconstrained optimization, *Comput. Appl. Math.* 241 (2013)19–29.

[2] N. Andrei, On three-term conjugate gradient algorithms for unconstrained optimization, *Applied Mathematics and Computation*, 219 (2013) 6316–6327.

[3] N. Andrei, An unconstrained optimization test functions collection, *Advanced Modeling and Optimization* 10 (2008) 147–161.

[4] E.M.L. Beale, A derivation of conjugate gradients, in: *Numerical Methods for Nonlinear Optimization*, F.A. Lootsma (Ed.), Academic Press, London, 1972, pp. 39–43.

[5] I. Bongartz, A.R. Conn, N. Gould and Ph.L.Toint, CUTE: constrained and unconstrained testing environments, *ACM Transactions on Mathematical Software*, 21 (1995) 123–160.

[6] Y.H. Dai and L.Z. Liao, New conjugacy conditions and related nonlinear conjugate gradient methods, *Applied Mathematics and Optimization* 43 (2001) 87–101.

[7] S.H. Deng and Z. Wan, An improved three-term conjugate gradient algorithm for solving unconstrained optimization problems, *Optimization* 64 (2015) 2679–2691.

[8] S.H. Deng and Z. Wan, A three-term conjugate gradient algorithm for large-scale unconstrained optimization problems, *Applied Numerical Mathematics* 92 (2015) 70–81.

[9] S.H. Deng, Z. Wan and X.H. Chen, An improved spectral conjugate gradient algorithm for nonconvex unconstrained optimization problems, *Journal of Optimization Theory and Applications* 157 (2013) 820–842.

[10] X.L. Dong, D.R. Han R. Ghanbari, et al. Some new three-term Hestenes-Stiefel conjugate gradient methods with affine combination, *Optimization* 66 (2017) 759–776.

[11] S.Q. Du and Y.Y. Chen, Global convergence of a modified spectral FR conjugate gradient method, *Applied Mathematics and Computation* 202 (2008) 766–770.

[12] W.W. Hager and H. Zhang, A survey of nonlinear conjugate gradient methods, *Pacific Journal of Optimization*, 2 (2006) 35–58.

[13] W.W Hager and H. Zhang, A new conjugate gradient method with guaranteed descent and an efficient line search, *SIAM Journal on Optimization*, 16 (2005) 170–192.

[14] S. Huang, Z. Wan and X.H. Chen, A new nonmonotone line search technique for unconstrained optimization, *Numerical Algorithms*, 68 (2015) 671–689.

[15] S. Huang, Z. Wan and J. Zhang, An extended nonmonotone line search technique for large-scale unconstrained optimization, *Journal of Computational and Applied Mathematics*, 330 (2018) 586–604.

[16] H. Kobayashi, Y. Narushima and H. Yabe, Descent three-term conjugate gradient methods based on secant conditions for unconstrained optimization, *Optimization Methods and Software*, 32 (2017) 1313–1329.

[17] T. Li and Z. Wan, New adaptive Barzilar-Borwein step size and its application in solving large scale optimization problems, *The ANZIAM Journal* 61 (2019) 76–98.

[18] Y.X. Li and Z. Wan, Bi-level programming approach to optimal strategy for VMI problems under random demand, *The ANZIAM Journal* 59 (2017) 247–270.

[19] Y. Narushima, H. Yabe and J.A. Ford, A three-term conjugate gradient method with sufficient descent property for unconstrained optimization, *SIAM Journal on Optimization* 21 (2011) 212–230.

[20] L. Nazareth, A conjugate direction algorithm without line search, *Journal of Optimization Theory and Applications* 23 (1977) 373–387.

[21] A. Perry, A modified conjugate gradient algorithm, *Operations Research* 26 (1978) 1073–1078.

[22] Z. Wan, J. Guo, J.J. Liu and W.Y. Liu, A modified spectral conjugate gradient projection method for signal recovery, *Signal Image and Video Processing* 12 (2018) 1455–1462.

[23] Z. Wan, C.M. Hu and Z.L. Yang, A spectral PRP conjugate gradient methods for nonconvex optimization problem based on modified line search, *Discrete and Continuous Dynamical Systems*, Series B, 16 (2011) 1157–1169.

[24] Z.X. We, G. Li and L. Qi, Global convergence of the Polak-Ribiere-Polyak conjugate gradient method with an Armijo-type inexact line search for nonconvex unconstrained optimization problems, *Mathematics of Computation* 77 (2008) 2173–2193.

[25] Y.L. Wu A modified three-term PRP conjugate gradient algorithm for optimization models, *Journal of Inequalities and Applications* 2017 (2017) 97.

[26] L. Zhang, W.J. Zhou and D.H. Li, Some descent three-term conjugate gradient methods and their global convergence, *Optimization Methods and Software* 22 (2007) 697–711.

[27] G. Zoutendijk, Nonlinear programming, computational methods in: *Integer and Nonlinear Programming*, J. Abadie(ed.), North-Holland, Amsterdam, 1970, pp. 37–86.

Manuscript received 3 April 2018
revised 6 September 2018
accepted for publication 13 November 2018

Songhai Deng
School of Mathematics and Statistics
Central South University, Changsha, 410083,China
E-mail address: dsonghai@163.com

Jing Lv
School of Mathematics and Statistics,
Central South University, Changsha, 410083,China
E-mail address: 321464219@qq.com

Zhong Wan
School of Mathematics and Statistics,
Central South University, Changsha, 410083, China
E-mail address: wanmath@163.com