COMMUN. MATH. SCI.
Vol. 21, No. 1, pp. 281–297

A GLOBALLY CONVERGENT DAI-LIAO CONJUGATE GRADIENT METHOD USING QUASI-NEWTON UPDATE FOR UNCONSTRAINED OPTIMIZATION*

YUTING CHEN†

Abstract. Using quasi-Newton update and acceleration scheme, a new Dai-Liao conjugate gradient method that does not need computing or storing any approximate Hessian matrix of the objective function is developed for unconstrained optimization. It is shown that the search direction derived from a modified Perry matrix not only possesses sufficient descent condition but also fulfills Dai-Liao conjugacy condition at each iteration. Under certain assumptions, we establish the global convergence of the proposed method for uniformly convex function and general function, respectively. The numerical results illustrate that the presented method can effectively improve the numerical performance and successfully solve the test problems with a maximum dimension of 100000.

Keywords. Conjugate gradient; unconstrained optimization; sufficient descent; conjugacy condition; global convergence.

AMS subject classifications. 65K05; 90C06; 90C30.

1. Introduction

Due to simple computation, low memory requirement and strong convergence property, conjugate gradient methods are widely used to solve unconstrained optimization with the following form:

$$\min f(x),\ x\in\mathbb{R}^n, \tag{1.1}$$

where $f:\ \mathbb{R}^n\to\mathbb{R}$ is a continuously differentiable function, bounded below and its gradient is denoted by $g(x)=\nabla f(x)$.

Starting from an initial guess $x_0\in\mathbb{R}^n$, the conjugate gradient methods generate a sequence $\{x_k\}$ as

$$x_{k+1}=x_k+\alpha_k d_k,\ k\geq 0, \tag{1.2}$$

where the step-length $\alpha_k>0$ satisfies some line search and the search direction d_k is determined by

$$d_0=-g_0,\ d_{k+1}=-g_{k+1}+\beta_k d_k,\ k\geq 0, \tag{1.3}$$

where $g_{k+1}=g(x_{k+1})$ and the scalar β_k is called the conjugate gradient parameter. Different choices for β_k result in different conjugate gradient methods.

One of the conjugate gradient methods' advantages is conjugacy, i.e., the generated search direction satisfies $d_{k+1}^{\mathrm{T}}y_k=0$, where $y_k=g_{k+1}-g_k$. Since practical numerical methods adopt inexact line search instead of exact ones, Dai and Liao [10] extended the classical conjugacy condition to

$$d_{k+1}^{\mathrm{T}}y_k=-t g_{k+1}^{\mathrm{T}}s_k, \tag{1.4}$$

*Received: April 07, 2021; Accepted (in revised form): May 29, 2022. Communicated by Wotao Yin.

†School of Mathematics, Jilin University and Tianyuan Mathematical Center in Northeast China, Changchun 130012, China (chenyuting1012@126.com).

where $t \geq 0$ is a scalar. Based on Dai-Liao conjugacy condition (1.4), Dai and Liao obtained $\beta_k^{\mathrm{DL}}(t)$ as follows:

$$\beta_k^{\mathrm{DL}}(t) = \frac{g_{k+1}^{\mathrm{T}} y_k}{d_k^{\mathrm{T}} y_k} - t \frac{g_{k+1}^{\mathrm{T}} s_k}{d_k^{\mathrm{T}} y_k}. \tag{1.5}$$

Many researchers have paid much attention to design the manners of computing the parameter t, which can greatly affect the numerical performance of Dai-Liao conjugate gradient methods. By letting the parameter $t = 2\frac{\|y_k\|^2}{s_k^{\mathrm{T}} y_k}$ and $t = \frac{\|y_k\|^2}{s_k^{\mathrm{T}} y_k}$, Hager and Zhang [14] and Dai and Kou [9] obtained CG_DESCENT method and DK method, respectively.

On the other hand, combining quasi-Newton update with conjugate gradient method, some researchers considered conjugate gradient method as a special type of quasi-Newton method. Based on this relation, Perry [17] introduced the following parameter β_k in (1.3)

$$\beta_k^{\mathrm{P}} = \frac{g_{k+1}^{\mathrm{T}} y_k}{d_k^{\mathrm{T}} y_k} - \frac{g_{k+1}^{\mathrm{T}} s_k}{d_k^{\mathrm{T}} y_k}. \tag{1.6}$$

By simple derivation, the Perry's search direction defined by (1.3) and (1.6) can be written as

$$d_{k+1}^{\mathrm{P}} = -Q_{k+1}^{\mathrm{P}} g_{k+1}, \tag{1.7}$$

where

$$Q_{k+1}^{\mathrm{P}} = I - \frac{s_k y_k^{\mathrm{T}}}{s_k^{\mathrm{T}} y_k} + \frac{s_k s_k^{\mathrm{T}}}{s_k^{\mathrm{T}} y_k}. \tag{1.8}$$

It is obvious that Perry conjugate gradient method can be considered as a special type of quasi-Newton method in which Q_{k+1}^{P} is used to approximate the inverse Hessian of the objective function. Since Q_{k+1}^{P} is nonsymmetric and does not satisfy the secant condition, (1.7) cannot be regarded as quasi-Newton direction from a strict point of view.

To overcome the above shortcomings, together with Dai-Liao conjugacy condition (1.4), Andrei [3, 5] suggested the following matrix Q_{k+1}^{N} to replace Q_{k+1}^{P} given in (1.8)

$$Q_{k+1}^{\mathrm{N}} = I - \frac{s_k y_k^{\mathrm{T}} - y_k s_k^{\mathrm{T}}}{s_k^{\mathrm{T}} y_k} + t \frac{s_k s_k^{\mathrm{T}}}{s_k^{\mathrm{T}} y_k}. \tag{1.9}$$

Yao et al. [18] introduced a three-term Dai-Liao conjugate gradient method, in which the following modified symmetric Perry matrix was used to approximate the inverse Hessian

$$Q_{k+1}^{\mathrm{MP}} = I - \omega_k \frac{s_k y_k^{\mathrm{T}} - y_k s_k^{\mathrm{T}}}{s_k^{\mathrm{T}} y_k} + \frac{s_k s_k^{\mathrm{T}}}{s_k^{\mathrm{T}} y_k}, \tag{1.10}$$

where ω_k was a positive parameter to be determined based on Dai-Liao conjugacy condition (1.4). The generated search directions in the above methods satisfy both the descent condition and Dai-Liao conjugacy condition. For relevant research see [2, 4, 6–8, 11, 19, 20].

Inspired by the above, we are interested in developing a new Dai-Liao conjugate gradient method based on the fact that conjugacy condition and quasi-Newton technique can improve the classical conjugate gradient method. The proper modification of Perry matrix can ensure the generated search direction possesses sufficient descent condition and Dai-Liao conjugacy condition. Meanwhile, there is no computation or storage of any approximate Hessian matrix of the objective function during the execution of the obtained method. For general function, we establish the global convergence of the proposed method under appropriate conditions, while most of the methods mentioned above (such as [2,3,5–8,11]) are only convergent for uniformly convex function.

The rest of this paper is organized as follows. In next section, we describe the framework of the new Dai-Liao conjugate gradient method using quasi-Newton update for unconstrained optimization. Global convergence results for uniformly convex function and general function will be established respectively under appropriate conditions in Section 3. Section 4 is devoted to numerical experiments and comparisons with some effective conjugate gradient algorithms. Conclusions are drawn in Section 5.

2. A new Dai-Liao conjugate gradient method

For the update matrices Q^{N}_{k+1} and Q^{MP}_{k+1} suggested in [3,5] and [18], it is reasonable to believe that the iterative update matrix can be expressed as

$$Q_{k+1}=I+t_1Q_2^{k+1}+t_2Q_1^{k+1},$$

where Q_2^{k+1} and Q_1^{k+1} are given rank 2 and rank 1 adjusted matrices, respectively. The search direction of the methods in [3,5,18] can satisfy the descent condition and Dai-Liao conjugacy condition. Nevertheless, the global convergence of the methods in [3,5] are established just for uniformly convex function. In order to obtain the global convergence for general function and simplify the computation appropriately, we let $t_1=t_2=1$ and propose the following simplified symmetric Perry matrix:

$$Q^{\mathrm{NP}}_{k+1}=I+Q_2^{k+1}+Q_1^{k+1}, \tag{2.1}$$

where

$$Q_2^{k+1}=-\frac{s_ky_k^{\mathrm{T}}-y_ks_k^{\mathrm{T}}}{s_k^{\mathrm{T}}y_k},\ Q_1^{k+1}=\frac{s_ks_k^{\mathrm{T}}}{s_k^{\mathrm{T}}y_k}. \tag{2.2}$$

The obtained method can be regarded as a special Dai-Liao conjugate gradient method, in which the search direction is generated by

$$d_{k+1}=-Q^{\mathrm{NP}}_{k+1}g_{k+1},\ k\geq 0. \tag{2.3}$$

Substituting (2.2) into (2.3), the generated search direction can be rewritten as a typical three-term conjugate gradient direction as follows:

$$d_{k+1}=-g_{k+1}+a_kd_k+b_ky_k, \tag{2.4}$$

where

$$a_k=\frac{g_{k+1}^{\mathrm{T}}y_k}{d_k^{\mathrm{T}}y_k}-\frac{g_{k+1}^{\mathrm{T}}s_k}{d_k^{\mathrm{T}}y_k}, \tag{2.5}$$

$$b_k=-\frac{g_{k+1}^{\mathrm{T}}s_k}{s_k^{\mathrm{T}}y_k}. \tag{2.6}$$

Note that if the exact line search is employed, then (2.5) and (2.6) yield $a_k = \frac{g_{k+1}^{\mathrm{T}} y_k}{d_k^{\mathrm{T}} y_k}$, $b_k = 0$, i.e.,

$$d_{k+1} = -g_{k+1} + \frac{g_{k+1}^{\mathrm{T}} y_k}{d_k^{\mathrm{T}} y_k} d_k = -g_{k+1} + \beta_k^{\mathrm{HS}} d_k,$$

which is exactly the classical HS conjugate gradient method [15].

Taking into consideration the acceleration scheme [1], the detailed steps of the new Dai-Liao conjugate gradient algorithm (NDLCG) can be formally stated as follows.

Algorithm 2.1 (NDLCG)

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 the Wolfe line search:

$$f(x_k + \alpha_k d_k) - f(x_k) \le \rho \alpha_k g_k^{\mathrm{T}} d_k, \tag{2.7}$$

and

$$g_{k+1}^{\mathrm{T}} d_k \ge \sigma g_k^{\mathrm{T}} d_k, \tag{2.8}$$

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

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 $\bar{b}_k = -\alpha_k d_k^{\mathrm{T}} y_k$;

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

Step 6. Compute d_{k+1} by (2.4), in which a_k and b_k are determined by (2.5) and (2.6), respectively.

Step 7. Set $k := k+1$ and go to Step 1.

REMARK 2.1. Step 3 corresponds to the acceleration scheme which is proposed by Andrei [1]. Specifically, the main idea is to modify the step-length α_k by means of a positive parameter ξ_k in a multiplicative manner. Numerical results in Section 4 show that the behavior of accelerated computational scheme outperforms the corresponding conjugate gradient method in such a way.

REMARK 2.2. The main computational cost lies in $g_{k+1}^{\mathrm{T}} s_k$, $g_{k+1}^{\mathrm{T}} y_k$ and $d_k^{\mathrm{T}} y_k$ in Step 5. For the objective function $f(x)$, where $x \in \mathbb{R}^n$, they cost $O(3n)$ operations to compute the values of a_k and b_k. Therefore, no additional computations or storage costs are required in the process.

3. Convergence analysis

We use the basic assumption on the objective function in the following.

Assumption A

(i) The level set $\Omega = \{x \in \mathbb{R}^n : f(x) \le f(x_0)\}$ is bounded, i.e., there exists a constant $B > 0$ such that $\|x\| \le B$, $\forall x \in \Omega$.

(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

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

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

$$\|g(x)\| \le \Gamma, \quad \forall x \in \Omega. \tag{3.2}$$

The descent and the conjugacy conditions of a proposed search direction are crucial in the convergence analysis for conjugate gradient methods. The search direction generated by NDLCG can possess both the sufficient descent condition and the Dai-Liao conjugacy condition, a concept to be discussed below. Since the line search satisfies the Wolfe line search (2.7) and (2.8), it follows that $s_k^{\mathrm{T}} y_k > 0$.

LEMMA 3.1. *Suppose that the search direction d_{k+1} is generated by* NDLCG. *Then the sufficient descent condition holds for all $k \ge 0$, i.e., there exists a positive constant c, such that*

$$g_{k+1}^{\mathrm{T}} d_{k+1} \le -c\|g_{k+1}\|^2. \tag{3.3}$$

Proof. Since $g_0^{\mathrm{T}} d_0 = -\|g_0\|^2$, the sufficient descent condition holds for $k = 0$. By direct computation, we get

$$\begin{aligned} g_{k+1}^{\mathrm{T}} d_{k+1} &= -\|g_{k+1}\|^2 + a_k g_{k+1}^{\mathrm{T}} d_k + b_k g_{k+1}^{\mathrm{T}} y_k \\ &= -\|g_{k+1}\|^2 + \frac{g_{k+1}^{\mathrm{T}} y_k g_{k+1}^{\mathrm{T}} s_k}{s_k^{\mathrm{T}} y_k} - \frac{(g_{k+1}^{\mathrm{T}} s_k)^2}{s_k^{\mathrm{T}} y_k} - \frac{g_{k+1}^{\mathrm{T}} s_k g_{k+1}^{\mathrm{T}} y_k}{s_k^{\mathrm{T}} y_k} \\ &= -\|g_{k+1}\|^2 - \frac{(g_{k+1}^{\mathrm{T}} s_k)^2}{s_k^{\mathrm{T}} y_k} \\ &\le -\|g_{k+1}\|^2, \end{aligned}$$

which means that the sufficient descent condition holds for $c = 1$, since $s_k^{\mathrm{T}} y_k > 0$. □

LEMMA 3.2. *Suppose that the search direction d_{k+1} is generated by* NDLCG. *Then d_{k+1} given by (2.4)-(2.6) satisfies the Dai-Liao conjugacy condition (1.4), i.e.,*

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

where the scalar $t > 0$ for all k.

Proof. By direct computation, we have

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

where $t = \left(1 + \frac{\|y_k\|^2}{s_k^{\mathrm{T}} y_k}\right) > 0$, since $s_k^{\mathrm{T}} y_k > 0$. □

Although the search direction d_k is always a descent direction, in order to get the convergence of NDLCG, we need to derive a lower bound for the step-length α_k.

LEMMA 3.3. *Suppose that Assumption* A *holds. Consider the conjugate gradient method* (1.2) *and* (2.4)-(2.6)*, where the step-length α_k is determined by the Wolfe line search* (2.7) *and* (2.8)*. Then α_k satisfies*

$$\alpha_k \geq \frac{(\sigma-1)g_k^{\mathrm{T}} d_k}{L\|d_k\|^2}, \tag{3.4}$$

where σ and L are positive constants in (2.8) *and* (3.1)*, respectively.*

Proof. Subtracting $g_k^{\mathrm{T}} d_k$ from both sides of (2.8) and using (3.1), we have

$$(\sigma-1)g_k^{\mathrm{T}} d_k \leq (g_{k+1}-g_k)^{\mathrm{T}} d_k = y_k^{\mathrm{T}} d_k \leq \|y_k\|\|s_k\| \leq \alpha_k L\|d_k\|^2.$$

Since d_k is a descent direction and $\sigma<1$, (3.4) follows immediately. □

The following lemma called Zoutendijk condition [21] plays an important role in the analysis of global convergence for conjugate gradient method.

LEMMA 3.4. *Suppose that Assumption* A *holds. Consider the conjugate gradient method* (1.2) *and* (2.4)-(2.6)*, where the step-length α_k is determined by the Wolfe line search* (2.7) *and* (2.8)*. Then*

$$\sum_{k=0}^{\infty} \frac{(g_k^{\mathrm{T}} d_k)^2}{\|d_k\|^2} < \infty. \tag{3.5}$$

Proof. Combining (2.7) with (3.4), we have

$$f(x_k) - f(x_k+\alpha_k d_k) \geq \frac{\rho(1-\sigma)(g_k^{\mathrm{T}} d_k)^2}{L\|d_k\|^2}.$$

By summing up both sides of the above inequality, and using the condition (i) of Assumption A, Zoutendijk condition (3.5) holds immediately. □

In what follows, we establish the global convergence theorem of NDLCG for uniformly convex function. If f is a uniformly convex function on Ω, then there exists a constant $\mu>0$ such that

$$(\nabla f(x)-\nabla f(y))^{\mathrm{T}}(x-y) \geq \mu\|x-y\|^2, \ \forall x,y \in \mathbb{N}. \tag{3.6}$$

LEMMA 3.5. *Suppose that Assumption* A *holds. Let $\{x_k\}$ and $\{d_k\}$ be generated by* NDLCG*. For given uniformly convex function f, the norms of the sequence $\{d_k\}$ is bounded above, i.e., there exists a constant $M>0$ such that*

$$\|d_k\| \leq M, \ \forall k \geq 0. \tag{3.7}$$

Proof. From (3.1), we have

$$\|y_k\| \leq L\|s_k\|. \tag{3.8}$$

From (3.6), we obtain

$$y_k^{\mathrm{T}} s_k \geq \mu\|s_k\|^2. \tag{3.9}$$

From Cauchy inequality and (3.9), it follows that $\mu\|s_k\|^2 \le y_k^{\mathrm{T}} s_k \le \|y_k\|\|s_k\|$, i.e.,

$$\mu\|s_k\| \le \|y_k\|. \tag{3.10}$$

By using (3.2) and (3.8)-(3.10) with (2.5) and (2.6) in (2.4), we get

$$\begin{aligned}
\|d_{k+1}\| &= \|-g_{k+1} + a_k d_k + b_k y_k\| \\
&= \|-g_{k+1} + \frac{g_{k+1}^{\mathrm{T}} y_k - g_{k+1}^{\mathrm{T}} s_k}{s_k^{\mathrm{T}} y_k} s_k - \frac{g_{k+1}^{\mathrm{T}} s_k}{s_k^{\mathrm{T}} y_k} y_k\| \\
&\le \|g_{k+1}\| + \frac{\|g_{k+1}\|\|y_k\| + \|g_{k+1}\|\|s_k\|}{s_k^{\mathrm{T}} y_k}\|s_k\| + \frac{\|g_{k+1}\|\|s_k\|}{s_k^{\mathrm{T}} y_k}\|y_k\| \\
&\le \Gamma + \frac{\Gamma L\|s_k\| + \Gamma\|s_k\|}{\mu\|s_k\|^2}\|s_k\| + \frac{\Gamma\|s_k\|}{\mu\|s_k\|^2} L\|s_k\| \\
&= \Gamma + \frac{\Gamma L + \Gamma}{\mu} + \frac{\Gamma L}{\mu} \equiv M.
\end{aligned}$$

The proof is completed. □

THEOREM 3.1. *Suppose that Assumption* A *holds. Let* $\{x_k\}$ *and* $\{d_k\}$ *be generated by* NDLCG. *For uniformly convex function, we have*

$$\lim_{k\to\infty} \|g_k\| = 0. \tag{3.11}$$

Proof. According to Lemmas 3.1, 3.4 and 3.5, we have

$$\infty > \sum_{k\ge 0} \frac{(g_k^{\mathrm{T}} d_k)^2}{\|d_k\|^2} \ge \frac{c^2}{M^2} \sum_{k\ge 0} \|g_k\|^4,$$

which deduces (3.11). □

Furthermore, in order to obtain the global convergence of NDLCG for general function, we need to make some modifications as follows:

(i) We make a nonnegative restriction on a_k as:

$$a_k^+ = \max\{a_k,\ 0\}, \tag{3.12}$$

where a_k is defined by (2.5). Then the search direction is generated by

$$d_{k+1} = -g_{k+1} + a_k^+ d_k + b_k y_k, \tag{3.13}$$

where a_k^+ and b_k are computed as (3.12) and (2.6), respectively.

(ii) In determination of the step-length α_k, we employ strong Wolfe line search instead of the Wolfe line search, which is given by (2.7) and

$$|g_{k+1}^{\mathrm{T}} d_k| \le -\sigma g_k^{\mathrm{T}} d_k. \tag{3.14}$$

For a general function f, we establish a weaker convergence result in the sense that

$$\liminf_{k\to\infty} \|g_k\| = 0. \tag{3.15}$$

Suppose that (3.15) does not hold, i.e., there is a positive constant γ such that

$$\|g_k\| > \gamma, \ \forall k \geq 0. \tag{3.16}$$

LEMMA 3.6. *Suppose that Assumption* A *holds. Let d_{k+1} be generated by* (3.13), *where a_k^+ and b_k are computed as* (3.12) *and* (2.6) *respectively, and the step-length α_k is determined by strong Wolfe line search* (2.7) *and* (3.14). *If* (3.16) *holds, then*

$$\sum_{k\geq 0} \|u_{k+1} - u_k\|^2 < \infty, \tag{3.17}$$

where $u_k = \dfrac{d_k}{\|d_k\|}$.

Proof. The definition of u_k is well-defined, since the search direction fulfills the sufficient descent condition, $d_k = 0$ implies $g_k = 0$ which contradicts (3.16). It follows from (3.13) that

$$\begin{aligned} \frac{d_{k+1}}{\|d_{k+1}\|} &= \frac{-g_{k+1}}{\|d_{k+1}\|} + a_k^+ \frac{d_k}{\|d_{k+1}\|} + b_k \frac{y_k}{\|d_{k+1}\|} \\ &= \frac{-g_{k+1} + b_k y_k}{\|d_{k+1}\|} + a_k^+ \frac{\|d_k\|}{\|d_{k+1}\|} \frac{d_k}{\|d_k\|}, \end{aligned} \tag{3.18}$$

which means

$$u_{k+1} = \omega_k + \nu_k u_k, \tag{3.19}$$

where

$$\omega_k = \frac{-g_{k+1} + b_k y_k}{\|d_{k+1}\|}, \tag{3.20}$$

$$\nu_k = a_k^+ \frac{\|d_k\|}{\|d_{k+1}\|} \geq 0. \tag{3.21}$$

Combining the identity $\|u_{k+1}\| = \|u_k\| = 1$ with (3.19), we have

$$\|\omega_k\| = \|u_{k+1} - \nu_k u_k\| = \|\nu_k u_{k+1} - u_k\|. \tag{3.22}$$

Using triangle inequality, (3.22) and $\nu_k \geq 0$, we get

$$\begin{aligned} \|u_{k+1} - u_k\| &\leq \|(1+\nu_k)u_{k+1} - (1+\nu_k)u_k\| \\ &\leq \|u_{k+1} - \nu_k u_k\| + \|\nu_k u_{k+1} - u_k\| \\ &= 2\|\omega_k\|. \end{aligned} \tag{3.23}$$

From strong Wolfe line search (3.14), we obtain

$$\left| \frac{g_{k+1}^{\mathrm{T}} d_k}{d_k^{\mathrm{T}} y_k} \right| \leq \frac{\sigma}{1-\sigma}. \tag{3.24}$$

By using the definition of (3.20) and (3.21) with (3.24), we have

$$\begin{aligned}\|\omega_k\| &\leq \frac{\|-g_{k+1}+b_k y_k\|}{\|d_{k+1}\|}\\ &\leq \frac{\|g_{k+1}\|+|b_k|\|y_k\|}{\|d_{k+1}\|}\\ &\leq \frac{\|g_{k+1}\|+\frac{\sigma}{1-\sigma}\left(\|g_{k+1}\|+\|g_{k+1}\|\frac{\|g_k\|}{\|g_{k+1}\|}\right)}{\|d_{k+1}\|}.\end{aligned}$$

It follows from (3.2) and (3.16) that

$$\begin{aligned}\|\omega_k\| \leq &\frac{\|g_{k+1}\|}{\|d_{k+1}\|}\left[1+\frac{\sigma}{1-\sigma}(1+\frac{\Gamma}{\gamma})\right]\\ \equiv &M_1\frac{\|g_{k+1}\|}{\|d_{k+1}\|}.\end{aligned} \tag{3.25}$$

From Lemmas 3.1 and 3.4, we obtain that

$$\infty \geq \sum_{k\geq 0}\frac{(g_{k+1}^{\mathrm{T}}d_{k+1})^2}{\|d_{k+1}\|^2} \geq \sum_{k\geq 0}\frac{c^2\|g_{k+1}\|^4}{\|d_{k+1}\|^2} \geq \sum_{k\geq 0}\frac{c^2\gamma^2\|g_{k+1}\|^2}{\|d_{k+1}\|^2}, \tag{3.26}$$

which, together with (3.23) and (3.25), yields (3.17). The proof is completed. □

Lemma 3.6 indicates that the generated unit direction u_k changes slowly and asymptotically. In the global convergence of conjugate gradient method for general functions, the following Property (∗) proposed by Gilbert and Nocedal [13] has been widely applied. Motivated by [13], we prove that Property (∗) holds for the Dai-Liao conjugate gradient method formed by (1.2) and (3.13), where a_k^+ and b_k are computed as (3.12) and (2.6) respectively, and the step-length α_k is determined by strong Wolfe line search (2.7) and (3.14).

Property (∗) Consider the conjugate gradient method (1.2) and (3.13), and suppose that

$$0<\gamma\leq\|g_k\|\leq\bar{\gamma},\ \forall k\geq 0. \tag{3.27}$$

Under this assumption, we say that the method has Property (∗) if there are constants $b>1$ and $\lambda>0$ such that for all k

$$|a_k|\leq b, \tag{3.28}$$

and

$$\|s_k\|\leq\lambda \Rightarrow |a_k|\leq\frac{1}{2b}. \tag{3.29}$$

LEMMA 3.7. *Suppose that Assumption* A *holds. Let d_{k+1} be generated by* (3.13)*, where a_k^+ and b_k are computed as* (3.12) *and* (2.6) *respectively, and the step-length α_k is determined by strong Wolfe line search* (2.7) *and* (3.14)*. Then* Property (∗) *holds.*

Proof. From the strong Wolfe line search (3.14) and the sufficient descent condition (3.3), we get

$$d_k^{\mathrm{T}}y_k \geq (\sigma-1)g_k^{\mathrm{T}}d_k \geq c(1-\sigma)\|g_k\|^2. \tag{3.30}$$

It follows from (3.27), (3.30) and the definition of a_k given by (2.5) that

$$\begin{aligned}|a_k| &= \left|\frac{g_{k+1}^{\mathrm{T}} y_k}{d_k^{\mathrm{T}} y_k} - \frac{g_{k+1}^{\mathrm{T}} s_k}{d_k^{\mathrm{T}} y_k}\right| \\ &\le \frac{\|g_{k+1}\|\|y_k\| + \|g_{k+1}\|\|s_k\|}{d_k^{\mathrm{T}} y_k} \\ &\le \frac{\|g_{k+1}\|(\|g_{k+1}\| + \|g_k\|) + \|g_{k+1}\|(\|x_{k+1}\| + \|x_k\|)}{c(1-\sigma)\|g_k\|^2} \\ &= \frac{2\bar{\gamma}^2 + 2B\bar{\gamma}}{c(1-\sigma)\gamma^2} \equiv b. \end{aligned} \tag{3.31}$$

It is worth mentioning that the value range of b exactly satisfies Property $(*)$, i.e., $b > 1$. We know from the previous discussion that $0 < \sigma < 1$, $\bar{\gamma} > \gamma > 0$, $B > 0$ and $c = 1$. Then

$$\frac{2\bar{\gamma}^2 + 2B\bar{\gamma}}{c(1-\sigma)\gamma^2} > \frac{2\gamma^2 + 2B\gamma}{2\gamma^2} = 1 + \frac{B}{\gamma} > 1.$$

Define

$$\lambda = \frac{c^2(1-\sigma)^2\gamma^4}{4\bar{\gamma}^2(\bar{\gamma}+B)(L+1)}. \tag{3.32}$$

If $\|s_k\| \le \lambda$, from the first inequality of (3.31), we obtain

$$\begin{aligned}|a_k| &\le \frac{\bar{\gamma}L\|s_k\| + \bar{\gamma}\|s_k\|}{c(1-\sigma)\|g_k\|^2} \\ &\le \frac{\bar{\gamma}L + \bar{\gamma}}{c(1-\sigma)\gamma^2}\|s_k\| \\ &\le \frac{\bar{\gamma}L + \bar{\gamma}}{c(1-\sigma)\gamma^2}\lambda = \frac{1}{2b}. \end{aligned} \tag{3.33}$$

The proof is completed. □

In next lemma, we will state that if the sequence $\{\|g_{k+1}\|\}$ is bounded away from zero, then a fraction of the steps cannot be too small. Let N denote the set of positive integers, for $\lambda > 0$, define

$$K^{\lambda} := \{i \in N:\ i \ge 1,\ \|s_i\| \ge \lambda\}, \tag{3.34}$$

that is, the set of integers corresponding to steps that are larger than λ. We will need to discuss groups of Δ consecutive iterates. For this purpose, let

$$K^{\lambda}_{k,\Delta} := \{i \in N:\ k \le i \le k+\Delta-1,\ \|s_i\| \ge \lambda\}. \tag{3.35}$$

Let $\left|K^{\lambda}_{k,\Delta}\right|$ denote the number of elements in $K^{\lambda}_{k,\Delta}$.

LEMMA 3.8. *Suppose that Assumption* A *holds. Let d_{k+1} be generated by* (3.13), *where a_k^+ and b_k are computed as* (3.12) *and* (2.6) *respectively, and the step-length α_k is determined by strong Wolfe line search* (2.7) *and* (3.14). *If* (3.16) *holds, then there is a constant $\lambda > 0$ such that for any $\Delta \in N$ and any index k_0, there exists a greater index $k \ge k_0$ such that*

$$\left|K^{\lambda}_{k,\Delta}\right| > \frac{\Delta}{2}. \tag{3.36}$$

Proof. Suppose by contradiction that for any $\lambda>0$, there exists $\Delta\in N$ and k_0 such that for any $k\geq k_0$, we have

$$\left|K^{\lambda}_{k,\Delta}\right|\leq\frac{\Delta}{2}. \tag{3.37}$$

It follows from the definition of b_k given by (2.6) with (3.2), (3.16), (3.24) and (3.37) that

$$\begin{aligned}\|b_k y_k\|&=\left|\frac{g_{k+1}^{\mathrm{T}}s_k}{s_k^{\mathrm{T}}y_k}\right|\|y_k\|\\&\leq\frac{\sigma}{1-\sigma}\left(\|g_{k+1}\|+\frac{\|g_k\|}{\|g_{k+1}\|}\|g_{k+1}\|\right)\\&\leq\frac{\sigma}{1-\sigma}\left(1+\frac{\Gamma}{\gamma}\right)\|g_{k+1}\|\equiv M_2\|g_{k+1}\|.\end{aligned} \tag{3.38}$$

Since the method fulfills Property (∗), which leads to the existence of $\lambda>0$ and $b>1$ such that (3.28) and (3.29) hold for all k. For this λ, let Δ and k_0 be given by (3.37), for any given index $l\geq k_0+1$, we have

$$\begin{aligned}\|d_{k+1}\|^2&\leq(a_k\|d_k\|+\|-g_{k+1}+b_k y_k\|)^2\\&\leq2a_k^2\|d_k\|^2+2\|-g_{k+1}+b_k y_k\|^2\\&\leq2a_k^2\|d_k\|^2+2(2\|g_{k+1}\|^2+2\|b_k y_k\|^2).\end{aligned} \tag{3.39}$$

By use of (3.38) and (3.39), we deduce that

$$\begin{aligned}\|d_{k+1}\|^2&\leq2a_k^2\|d_k\|^2+4(1+M_2^2)\|g_{k+1}\|^2\\&\leq2a_k^2\|d_k\|^2+4(1+M_2^2)\bar{\gamma}^2\equiv2a_k^2\|d_k\|^2+M_3.\end{aligned} \tag{3.40}$$

The remaining proof is the same as Lemma 4.2 in [13], and hence the details are omitted. The proof is then completed. □

Based on Lemmas 3.6-3.8, we establish the global convergence of the proposed Dai-Liao conjugate gradient method for general function.

THEOREM 3.2. *Suppose that Assumption* A *holds. Let d_{k+1} be generated by* (3.13), *where a_k^+ and b_k are computed as* (3.12) *and* (2.6) *respectively, and the step-length α_k is determined by strong Wolfe line search* (2.7) *and* (3.14). *Then* (3.15) *holds.*

Proof. We proceed by contradiction. Assume that (3.15) does not hold, which means that the condition (3.16) holds and the Lemmas 3.6-3.8 further hold. Together with Assumption A, we obtain a contradiction similarly to the proof of Theorem 4.3 in [13] and omit the details here. The proof is then completed. □

4. Numerical experiments

In this section, we report some numerical results. All codes are written in Matlab R2013a and ran on PC with 1.80 GHz CPU processor and 8.00 GB RAM memory. The iteration is terminated by the following condition

$$\|g_k\|\leq\varepsilon\quad\text{or}\quad|f(x_{k+1})-f(x_k)|\leq\varepsilon\max\{1.0,|f(x_k)|\}. \tag{4.1}$$

The relevant parameters are specified as follows. In (4.1), $\varepsilon=10^{-6}$. In the Wolfe line search, $\rho=0.0001$, $\sigma=0.8$. The other parameters are set as default.

We select a number of 10 unconstrained optimization test problems from [16]. For each problem, we report 15 numerical experiments with different dimensions from 5000 to 100000. That is, we need to consider a set of 150 large-scale test problems, see Table 4.1 for details. All algorithms share the same stopping criteria and initial point.

No.	Prob	dim
1.	Allgower function	5000, 6000, ..., 9000, 10000, 20000, ..., 90000, 100000
2.	Penalty function I	5000, 6000, ..., 9000, 10000, 20000, ..., 90000, 100000
3.	Boundary value function	5000, 6000, ..., 9000, 10000, 20000, ..., 90000, 100000
4.	Schittkowski function 302	5000, 6000, ..., 9000, 10000, 20000, ..., 90000, 100000
5.	Yang tridiagonal function	5000, 6000, ..., 9000, 10000, 20000, ..., 90000, 100000
6.	Variable dimension function	5000, 6000, ..., 9000, 10000, 20000, ..., 90000, 100000
7.	Broyden tridiagonal function	5000, 6000, ..., 9000, 10000, 20000, ..., 90000, 100000
8.	Extended Rosenbrock function	5000, 6000, ..., 9000, 10000, 20000, ..., 90000, 100000
9.	Generalized Rosenbrock function	5000, 6000, ..., 9000, 10000, 20000, ..., 90000, 100000
10.	Extended Powell singular function	5000, 6000, ..., 9000, 10000, 20000, ..., 90000, 100000

TABLE 4.1. *The test problems and their dimensions.*

The following numerical experiments mainly include two parts.

Part 1. The comparisons between NDLCG and NDLCG without the acceleration scheme that is called NDLCG− for simplicity.

To compare the performance of different methods more clearly, we employ the profiles introduced by Dolan and Moré [12]. The left axis gives the percentage of the test problems for which a method is the fastest (efficiency), while the right side gives the percentage of the test problems that are successfully solved by each of the methods (robustness). In a performance profile plot, the top curve corresponds to the method that solves the most problems in a time that is within a factor of the best time.

Figures 4.1-4.4 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. For example, when comparing NDLCG with NDLCG− subject to the number of iterations (i.e., Figure 4.1), we see that NDLCG is the top performer. NDLCG is better in 33 problems (i.e., it acheived the minimum number of iterations in 33 problems), NDLCG− is better in only 2 problems, and they have the same number of iterations in 115 problems. The data with the same meaning are shown in the bottom right of Figures 4.2-4.4. Having that in view, we claim that the acceleration strategy can improve the numerical performance of conjugate gradient method in a way.

Part 2. The comparisons between NDLCG, CD_DESCENT [14], DK [9], THREECG [3] and TTCG [5].

CD_DESCENT and DK are recognized as two of the most popular conjugate gradient methods that have good numerical performance. THREECG and TTCG are both three-term conjugate gradient methods which have similar structure with NDLCG. For the sake of fairness, combining with acceleration scheme and by means of the Wolfe line search, we study the numerical comparisons of the five conjugate gradient methods in the form

$$d_{k+1} = -g_{k+1} + a_k d_k + b_k y_k.$$

· NDLCG: a_k and b_k are determined by (2.5) and (2.6), respectively.

· CD_DESCENT:

$$a_k = \frac{g_{k+1}^{\mathrm{T}} y_k}{d_k^{\mathrm{T}} y_k} - 2\frac{\|y_k\|^2}{s_k^{\mathrm{T}} y_k}\frac{g_{k+1}^{\mathrm{T}} s_k}{d_k^{\mathrm{T}} y_k},\ b_k = 0.$$

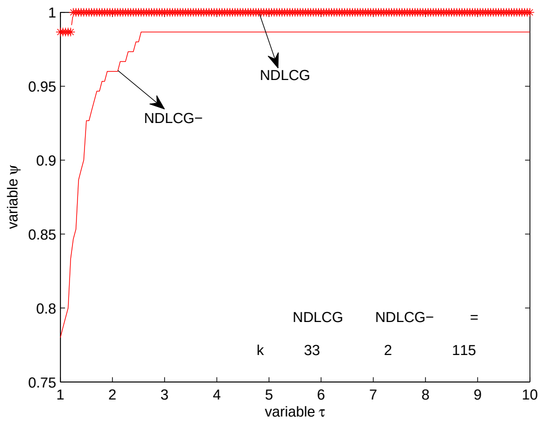

FIG. 4.1. *The number of iterations (k).*

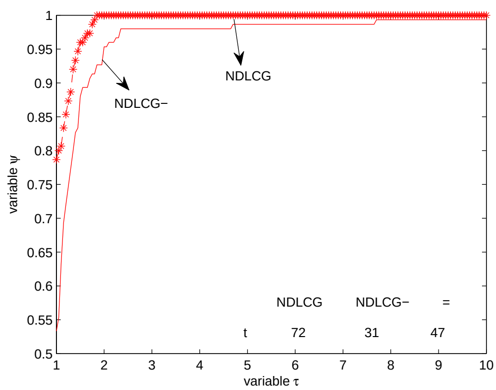

FIG. 4.2. *The CPU time (t).*

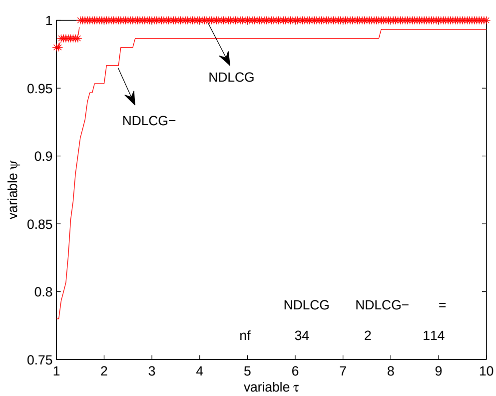

FIG. 4.3. *The number of function evaluations (nf).*

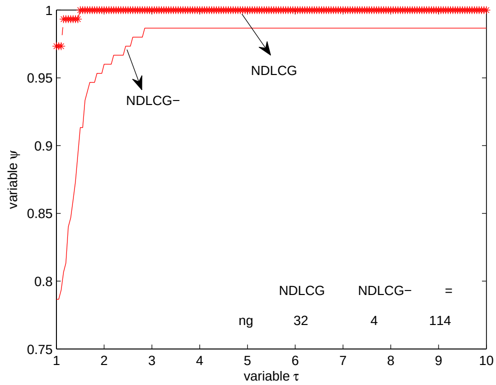

FIG. 4.4. *The number of gradient evaluations (ng).*

· DK:

$$a_k = \frac{g_{k+1}^{\mathrm{T}} y_k}{d_k^{\mathrm{T}} y_k} - \frac{\|y_k\|^2}{s_k^{\mathrm{T}} y_k} \frac{g_{k+1}^{\mathrm{T}} s_k}{d_k^{\mathrm{T}} y_k}, \ b_k = 0.$$

· THREECG:

$$a_k = \frac{g_{k+1}^{\mathrm{T}} y_k}{d_k^{\mathrm{T}} y_k} - \left(1 + \frac{\|y_k\|^2}{s_k^{\mathrm{T}} y_k}\right) \frac{g_{k+1}^{\mathrm{T}} s_k}{d_k^{\mathrm{T}} y_k}, \ b_k = -\frac{g_{k+1}^{\mathrm{T}} s_k}{s_k^{\mathrm{T}} y_k}.$$

· TTCG:

$$a_k = \frac{g_{k+1}^{\mathrm{T}} y_k}{d_k^{\mathrm{T}} y_k} - \left(1 + 2\frac{\|y_k\|^2}{s_k^{\mathrm{T}} y_k}\right) \frac{g_{k+1}^{\mathrm{T}} s_k}{d_k^{\mathrm{T}} y_k}, \ b_k = -\frac{g_{k+1}^{\mathrm{T}} s_k}{s_k^{\mathrm{T}} y_k}.$$

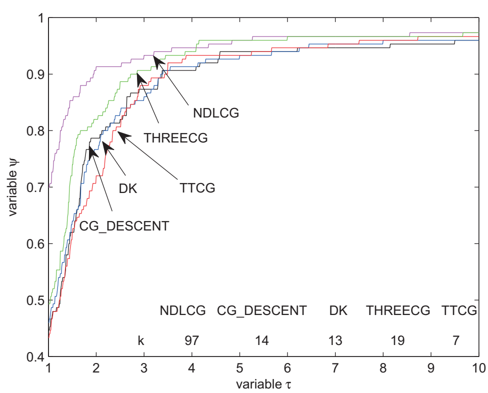

FIG. 4.5. *The number of iterations (k).*

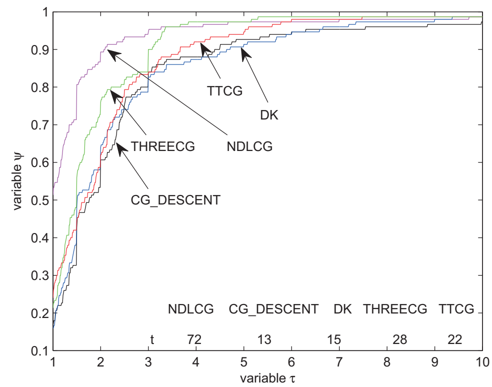

FIG. 4.6. *The CPU time (t).*

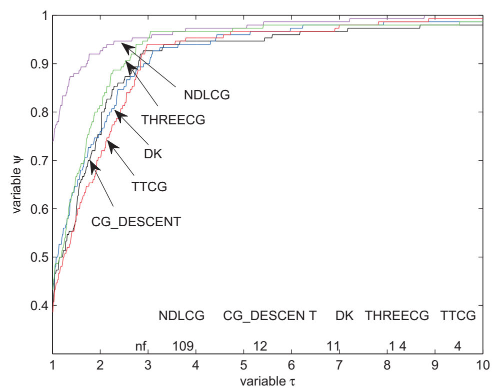

FIG. 4.7. *The number of function evaluations (nf).*

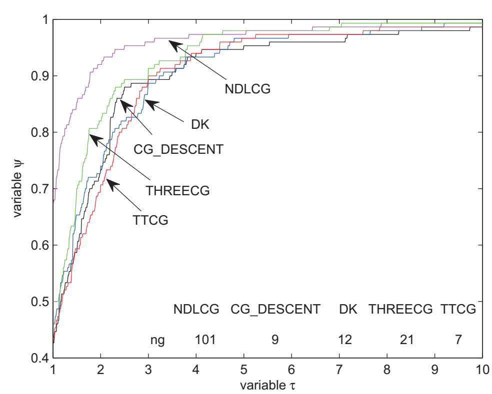

FIG. 4.8. *The number of gradient evaluations (ng).*

From Figure 4.5, subject to the number of iterations, NDLCG outperforms in 97 problems, THREECG outperforms in 19 problems, CG_DESCENT and DK have similar performance that outperforms in 14 problems and 13 problems respectively, while TTCG only outperforms in 7 problems. The data shown in the bottom right of Figures 4.6-4.8 have the same meaning.

As shown in Figure 4.6, with respect to the CPU time, CG_DESCENT, DK and TTCG have similar performance and all worse than NDLCG and THREECG slightly. It is obvious that the curve "NDLCG" is the top performer.

Similar with the results of Figures 4.5-4.6, Figures 4.7-4.8 give the profiles with respect to the number of function evaluations and gradient evaluations respectively, from which we can find NDLCG is slightly better than the other four methods. Although the four methods compared with NDLCG perform slightly different, their performances are quite close in general.

The numerical results demonstrate that NDLCG not only dominates THREECG and TTCG, but also CG_DESCENT and DK with acceleration scheme, which were regarded as some of the most effective conjugate gradient methods. Moreover, NDLCG also has high efficiency and robust numerical performance even if τ is small (see $\tau \in [1,3]$). To conclude, the proposed NDLCG in this paper is competitive and suitable for solving large-scale problems.

5. Conclusions

Conjugate gradient method represents an important class of optimization algorithms characterized by strong convergence and nice numerical performance. In this paper, we focus on combining the conjugate gradient method with quasi-Newton technology and Dai-Liao conjugacy condition, and suggest a new Dai-Liao conjugate gradient method with acceleration scheme for solving unconstrained optimization. The generated search direction is a linear combination of $-g_{k+1}$, d_k and y_k with the form $d_{k+1} = -g_{k+1} + a_k d_k + b_k y_k$, where the choices of two parameters a_k and b_k ensure that d_{k+1} satisfies sufficient descent and Dai-Liao conjugacy conditions. Under suitable assumptions, the global convergence of the obtained method for uniformly convex function and general function are established, respectively. Furthermore, we compare the numerical performance of the presented method against four other conjugate gradient methods, the numerical results indicate that the proposed method is always efficient as the dimension of problem increases.

Future research includes developing better choices of parameters to improve the numerical performance of the conjugate gradient methods, and extending the convergence results to other nonlinear conjugate gradient methods in a very economical fashion.

REFERENCES

[1] N. Andrei, *Acceleration of conjugate gradient algorithms for unconstrained optimization*, Appl. Math. Comput., 213:361–369, 2009. 2, 2.1

[2] N. Andrei, *A new three-term conjugate gradient algorithm for unconstrained optimization*, Numer. Algor., 68:305–321, 2015. 1

[3] N. Andrei, *A simple three-term conjugate gradient algorithm for unconstrained optimization*, J. Comput. Appl. Math., 241:19–29, 2013. 1, 1, 2, 4

[4] N. Andrei, *Nonlinear Conjugate Gradient Methods for Unconstrained Optimization*, Springer Optimization and Its Applications, Springer, 158, 2020. 1

[5] N. Andrei, *On three-term conjugate gradient algorithms for unconstrained optimization*, Appl. Math. Comput., 219:6316–6327, 2013. 1, 1, 2, 4

[6] S. Babaie-Kafaki and R. Ghanbari, *A descent family of Dai-Liao conjugate gradient methods*, Optim. Methods Softw., 29:583–591, 2014. 1

[7] S. Babaie-Kafaki and R. Ghanbari, *Descent symmetrization of the Dai-Liao conjugate gradient method*, Asia Pac. J. Oper. Res., 33:1650008, 2016. 1
[8] Y.T. Chen and Y.T. Yang, *A three-term conjugate gradient algorithm using subspace for large-scale unconstrained optimization*, Commun. Math. Sci., 18:1179–1190, 2020. 1
[9] Y.H. Dai and C.X. Kou, *A nonlinear conjugate gradient algorithm with an optimal property and an improved Wolfe line search*, SIAM J. Optim., 23:296–320, 2013. 1, 4
[10] Y.H. Dai and L.Z. Liao, *New conjugacy conditions and related nonlinear conjugate gradient methods*, Appl. Math. Optim., 43:87–101, 2001. 1
[11] S.H. Deng and Z. Wan, *A three-term conjugate gradient algorithm for large-scale unconstrained optimization problems*, Appl. Numer. Math., 92:70–81, 2015. 1
[12] E.D. Dolan and J.J. Moré, *Benchmarking optimization software with performance profiles*, Math. Program., 91:201–213, 2002. 4
[13] J.C. Gilbert and J. Nocedal, *Global convergence properties of conjugate gradient methods for optimization*, SIAM J. Optim., 2:21–42, 1992. 3, 3, 3
[14] W.W. Hager and H.C. Zhang, *A new conjugate gradient method with guaranteed descent and an efficient line search*, SIAM J. Optim., 16:170–192, 2005. 1, 4
[15] M.R. Hestenes and E.L. Stiefel, *Methods of conjugate gradient for solving linear systems*, J. Res. Nat. Bur. Standards, 49:409–436, 1952. 2
[16] J.J. Moré, B.S. Garbow, and K.E. Hillstrom, *Testing unconstrained optimization software*, ACM Trans. Math. Softw., 7:17–41, 1981. 4
[17] A. Perry, *Technical note-A modified conjugate gradient algorithm*, Oper. Res., 26:1073–1078, 1978. 1
[18] S.W. Yao, Q.L. Feng, L. Li, and J.Q. Xu, *A class of globally convergent three-term Dai-Liao conjugate gradient methods*, Appl. Numer. Math., 151:354–366, 2020. 1, 2
[19] S.W. Yao and L.S. Ning, *An adaptive three-term conjugate gradient method based on self-scaling memoryless BFGS matrix*, J. Comput. Appl. Math., 332:72–85, 2018. 1
[20] K.K. Zhang, H.W. Liu, and Z.X. Liu, *A new Dai-Liao conjugate gradient method with optimal parameter choice*, Numer. Funct. Anal. Optim., 40:194–215, 2019. 1
[21] G. Zoutenijk, *Nonlinear programming, computational methods*, in J. Abadie (ed.), Integer and Nonlinear Programming, North-Holland, Amsterdam, 37–86, 1970. 3