Computational Optimization and Applications
https://doi.org/10.1007/s10589-019-00143-4

An improved Dai–Kou conjugate gradient algorithm for unconstrained optimization

Zexian Liu[1,2] · Hongwei Liu[1] · Yu-Hong Dai[2]

Received: 25 November 2018

Abstract
It is gradually accepted that the loss of orthogonality of the gradients in a conjugate gradient algorithm may decelerate the convergence rate to some extent. The Dai–Kou conjugate gradient algorithm (SIAM J Optim 23(1):296–320, 2013), called CGOPT, has attracted many researchers' attentions due to its numerical efficiency. In this paper, we present an improved Dai–Kou conjugate gradient algorithm for unconstrained optimization, which only consists of two kinds of iterations. In the improved Dai–Kou conjugate gradient algorithm, we develop a new quasi-Newton method to improve the orthogonality by solving the subproblem in the subspace and design a modified strategy for the choice of the initial stepsize for improving the numerical performance. The global convergence of the improved Dai–Kou conjugate gradient algorithm is established without the strict assumptions in the convergence analysis of other limited memory conjugate gradient methods. Some numerical results suggest that the improved Dai–Kou conjugate gradient algorithm (CGOPT (2.0)) yields a tremendous improvement over the original Dai–Kou CG algorithm (CGOPT (1.0)) and is slightly superior to the latest limited memory conjugate gradient software package CG_DESCENT (6.8) developed by Hager and Zhang (SIAM J Optim 23(4):2150–2168, 2013) for the CUTEr library.

Keywords Conjugate gradient algorithm · Limited memory · Quasi-Newton method · Preconditioned conjugate gradient algorithm · Global convergence

Mathematics Subject Classification 90C06 · 90C26 · 65Y20

✉ Yu-Hong Dai
dyh@lsec.cc.ac.cn

Extended author information available on the last page of the article

Published online: 02 November 2019

1 Introduction

Consider the following unconstrained optimization problem

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

where $f:\mathbb{R}^n\to\mathbb{R}$ is continuously differentiable and its gradient is denoted by g.

Throughout this paper, $g_k = g(x_k)$, $f_k = f(x_k)$, $s_{k-1} = x_k - x_{k-1}$, $y_{k-1} = g_k - g_{k-1}$ and $\lambda_{\max}(\cdot)$ represents the maximum eigenvalue function. If $x\in\mathbb{R}^n$ and $\mathcal{S}\subset\mathbb{R}^n$, then $\operatorname{dist}\{x,\mathcal{S}\} = \inf\{\|y-x\|, y\in\mathcal{S}\}$.

Conjugate gradient (CG) algorithms are a class of powerful algorithms for large scale unconstrained optimization. CG algorithms take the following form

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

where α_k is the stepsize and d_k is the search direction given by

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

where β_k is usually called conjugate parameter.

Different choices of β_k lead to different CG algorithms. Some well-known formulae for β_k are called the Fletcher–Reeves (FR) [1], Hestenes–Stiefel (HS) [2], Polak–Ribière–Polyak (PRP) [3,4] and Dai–Yuan (DY) [7] formulae, and are given by

$$\beta_k^{FR} = \frac{\|g_{k+1}\|^2}{\|g_k\|^2}, \quad \beta_k^{HS} = \frac{g_{k+1}^T y_k}{d_k^T y_k}, \quad \beta_k^{PRP} = \frac{g_{k+1}^T y_k}{\|g_k\|^2}, \quad \beta_k^{DY} = \frac{\|g_{k+1}\|^2}{d_k^T y_k}.$$

In 2005, Hager and Zhang [9] proposed an efficient CG algorithm (CG_DESCENT) with

$$\beta_k^{HZ} = \frac{g_{k+1}^T y_k}{d_k^T y_k} - \theta\frac{\|y_k\|^2}{d_k^T y_k}\frac{g_{k+1}^T d_k}{d_k^T y_k}, \tag{4}$$

where θ is a parameter, and established the convergence of CG_DESCENT with the standard Wolfe line search. And the numerical results in [9,20] indicated that CG_DESCENT with the approximate Wolfe line search (AWolfe line search):

$$\sigma g_k^T d_k \leq g(x_k+\alpha_k d_k)^T d_k \leq (2\delta-1) g_k^T d_k,$$

where $0<\delta<0.5$ and $\delta\leq\sigma<1$, is very efficient.

By taking a multiple of the memoryless BFGS direction of Perry [6] and Shanno [5] and projecting it into the manifold $\{-g_{k+1}+sd_k : s\in\mathbb{R}\}$, Dai and Kou [8] recently developed a family of CG algorithms (CGOPT). We also call it Dai–Kou CG algorithms for short) with the improved Wolfe line search:

$$f(x_k+\alpha_k d_k) \leq f(x_k) + \min\left\{\epsilon|f(x_k)|, \delta\alpha_k g_k^T d_k + \bar{\eta}_k\right\}, \tag{5}$$

$$g(x_k+\alpha_k d_k)^T d_k \geq \sigma g_k^T d_k, \tag{6}$$

where $0 < \epsilon, 0 < \delta < \sigma < 1, 0 < \bar{\eta}_k$ and $\sum_{k\geq 0} \bar{\eta}_k < +\infty$, and the numerical results in [8] suggested that CGOPT with the following parameter:

$$\beta_k^{DK} = \frac{g_{k+1}^T y_k}{d_k^T y_k} - \frac{\|y_k\|^2}{d_k^T y_k} \frac{g_{k+1}^T d_k}{d_k^T y_k} \tag{7}$$

is the most efficient. CG_DESCENT and CGOPT are both popular and quite efficient CG software packages. Some recent advances about CG method can be found in [10–16].

Recently, Hager and Zhang [16] observed that for some ill-conditioned strictly convex quadratic problems, CG method with the exact line search might converge very slowly, while the unscaled limited memory BFGS algorithm (L-BFGS) [18,19] with the same line search converges quickly, although these two methods should yield exactly the same iterates in theory. They also monitored that the orthogonality of the successive gradients loses quickly during the iterations of CG method, while this is not true for the L-BFGS method. Based on the above observations, Hager and Zhang [16] thought that the slow convergence rate of CG method might be caused by the loss of orthogonality, first combined the limited memory technique with CG algorithm and presented a limited memory CG method (CG_DESCENT (6.0)), which can be regarded as a preconditioned CG method with the following three preconditioners:

$$P_k = I, \quad P_k = Z_k \hat{B}_{k+1}^{-1} Z_k^T, \quad P_k = Z_k \hat{B}_{k+1}^{-1} Z_k^T + \sigma_k \overline{Z}_k \overline{Z}_k^T, \tag{8}$$

where σ_k is given by (4.2) of [16], $\hat{B}_{k+1}$ is an approximation to the Hessian matrix of f at the subspace spanned by the previous search directions, and Z_k and $\overline{Z}_k$ are the matrices whose columns are the orthogonal basis for the above subspace and its complement, respectively. And the convergence of the limited memory CG method [16] with the standard Wolfe line search is established by imposing the following assumptions on the preconditioners (8):

$$\|P_k\| \leq \gamma_0, \qquad g_{k+1}^T P_k g_{k+1} \geq \gamma_1 \|g_{k+1}\|^2, \qquad d_k^T P_k^{-1} d_k \geq \gamma_2 \|d_k\|^2, \tag{9}$$

where $\gamma_0 > 0$, $\gamma_1 > 0$ and $\gamma_2 > 0$. The numerical results in [16] suggested that CG_DESCENT (6.0) has a significant improvement over the memoryless version CG_DESCENT (5.3).

Though the limited memory CG method [16] is surprisingly effective, there are still some drawbacks: (i) CG_DESCENT (6.0) with the AWolfe line search has illustrated very nice numerical performance, but there is no guarantee for the convergence of CG_DESCENT with the AWolfe line search [20]. While CG_DESCENT (6.0) with the standard Wolfe line search is globally convergent, but it performs significantly worse than CG_DESCENT (6.0) with the AWolfe line search; (ii) The assumptions (9), which are imposed on the preconditioners in the convergence analysis are relatively strict and not easy to verify in practice; (iii) The limited memory CG method [16] consists three kinds of iterations corresponding to the three preconditioners (8), which makes the limited memory CG method complicated.

To deal with the above three drawbacks in the limited memory CG method [16], we present an improved Dai–Kou CG algorithm for unconstrained optimization in this paper, which only consists of two kinds of iterations. In the improved Dai–Kou CG algorithm, in order to improve the orthogonality, we develop a new quasi-Newton method for solving the subproblem in the subspace spanned by the previous search directions. Motivated by the choice of the initial stepsize in [8], we also design a modified strategy for choosing the initial stepsize. The convergence of the improved Dai–Kou CG algorithm is established without the assumptions (9). Some numerical results are presented, which indicate that the improved Dai–Kou CG algorithm not only has a tremendous improvement over the original Dai–Kou CG algorithm but also outperforms the latest limited memory CG software package CG_DESCENT (6.8) [16].

The rest of the paper is organized as follows. In the next section, we develop a new quasi-Newton method in the subspace spanned by some previous search directions for improving the orthogonality, design a modified strategy for choosing the initial stepsize and present an improved Dai–Kou CG algorithm for unconstrained optimization. In Sect. 3, we establish the global convergence of the improved Dai–Kou CG algorithm without the assumptions (9). In Sect. 4, some numerical experiments are conducted to examine the effectiveness of the improved Dai–Kou CG algorithm. Conclusions are made in the last section.

2 The improved Dai–Kou CG algorithm

In the section, we develop a new quasi-Newton method for the subproblem in the subspace to improve the orthogonality, in which the search direction will be always transformed to the full space $\mathbb{R}^n$. A modified strategy for the choice of the initial stepsize is also designed later. We finally describe an improved Dai–Kou CG algorithm in detail, which only consists of two kinds of iterations.

We first consider the preconditioned version of CG algorithm (3) with (7). Suppose that P_k is a symmetric and positive definite preconditioner, the search direction of the preconditioned CG algorithm (3) with (7) is

$$d_{k+1} = -P_k g_{k+1} + \beta_k^{PDK} d_k, \tag{10}$$

where

$$\beta_k^{PDK} = \frac{g_{k+1}^T P_k y_k}{d_k^T y_k} - \frac{y_k^T P_k y_k}{d_k^T y_k} \frac{g_{k+1}^T d_k}{d_k^T y_k}. \tag{11}$$

Clearly, if $P_k = I$, then the search direction (10) reduces to the original CG direction (3) with (7). In order to establish the convergence and improve the numerical performance, we take the following truncated form:

$$d_{k+1} = -P_k g_{k+1} + \beta_k^{PDK+} d_k, \tag{12}$$

where

$$\beta_k^{PDK+} = \max\left\{\beta_k^{PDK}, \eta_k\right\}, \quad \eta_k = -\eta\frac{\left|g_{k+1}^T d_k\right|}{d_k^T P_k^{-1} d_k}, \tag{13}$$

where $\eta \in [0, 1)$ and P_k^{-1} denotes the pseudoinverse of P_k. It is noted that η_k in (13) is originated from

$$\eta_k = -\eta\frac{\left|g_{k+1}^T d_k\right|}{d_k^T d_k}, \tag{14}$$

which is based on the scheme $\eta_k = \eta\frac{g_{k+1}^T d_k}{d_k^T d_k}$ suggested by Dai and Kou [8]. The idea behind is to give more opportunities for the case of $\beta_k^{PDK+} = \beta_k^{PDK}$.

The improved Dai–Kou CG algorithm mainly consists of the following two kinds of iterations:

(1) Standard CG iteration

The search direction in the standard CG iteration corresponds to (12) with $P_k = I$, namely,

$$d_{k+1} = -g_{k+1} + \max\left\{\beta_k^{DK}, -\eta\frac{\left|g_{k+1}^T d_k\right|}{d_k^T d_k}\right\} d_k. \tag{15}$$

The standard CG iteration will be interrupted if the current gradient g_k is not approximately orthogonal to the following subspace:

$$\mathcal{S}_k = span\left\{d_{k-1}, d_{k-2}, \ldots, d_{k-m}\right\},$$

where m is a positive integer, and then the iteration turns to the following subspace iteration.

(2) Subspace iteration

When the orthogonality of the sequence of gradients in the CG algorithm is lost, the iteration switches from the standard CG iteration to the subspace iteration. In the subspace iteration described in Sect. 2.1, a new quasi-Newton method in the subspace $\mathcal{S}_k$ is developed to improve the orthogonality, in which the search direction will be always transformed to the full space $\mathbb{R}^n$. The main part of the resulting search direction can be regarded as a preconditioned CG direction.

If the orthogonality is improved, the iteration will depart the subspace, and the standard CG iteration (15) is evoked immediately. While the limited memory CG algorithm [16], which consists of three kinds of iterations: standard CG iteration, subspace iteration and a special preconditioned CG iteration with the complicated preconditioner corresponding to the third term in (8), first performs the special preconditioned CG iteration.

2.1 A new quasi-Newton method in the subspace for improving the orthogonality

Let $S_k \in \mathbb{R}^{n\times m}$ be such a matrix whose columns are $d_{k-1}, d_{k-2}, \ldots, d_{k-m}$. We suppose that the columns of S_k are linearly independent. It is also observed that the case of linear dependence rarely occurs. Let the QR factorization of S_k be $S_k = Z_k \bar{R}_k$, where the columns of $Z_k \in \mathbb{R}^{n\times m}$ form the normal orthogonal basis of $\mathcal{S}_k$ and $\bar{R}_k \in \mathbb{R}^{m\times m}$ is the upper triangular matrix with positive diagonal entries.

If g_k is nearly in the subspace $\mathcal{S}_k$, then CG algorithm has lost the orthogonality which can be detected by the distance of the current gradient g_k and the subspace $\mathcal{S}_k$:

$$\operatorname{dist}\{g_k, \mathcal{S}_k\} \leq \widetilde{\eta}_0 \|g_k\|, \tag{16}$$

where $0 < \widetilde{\eta}_0 < 1$ is small. Since the columns of Z_k form the normal orthogonal basis of $\mathcal{S}_k$, it is not difficult to obtain from the definition of $\operatorname{dist}\{g_k, \mathcal{S}_k\}$ that (16) can be written as

$$\left(1 - \widetilde{\eta}_0^2\right) \|g_k\|^2 \leq \left\| Z_k^T g_k \right\|^2. \tag{17}$$

The inequality (17) implies that the trial search direction (15) almost belongs to the subspace $\mathcal{S}_k$. In the case, it seems that it is better to optimize in the subspace $\mathcal{S}_k$ than to continue the iteration in the full space $\mathbb{R}^n$, since the subspace $\mathcal{S}_k$ has not been fully utilized and the dimension of the subspace $\mathcal{S}_k$ is usually small. As a result, we temporarily terminate the standard CG iteration and turn to optimize the objective function over $\mathcal{S}_k$:

$$\min_{z\in\mathcal{S}_k} \; f(x_k + z). \tag{18}$$

If the gradient g_{k+1} becomes sufficiently orthogonal to the subspace, which can be measured by $\operatorname{dist}\{g_{k+1}, \mathcal{S}_k\} \geq \widetilde{\eta}_1 \|g_{k+1}\|$, where $0 < \widetilde{\eta}_0 < \widetilde{\eta}_1 < 1$, then the iteration will leave the subspace $\mathcal{S}_k$. Similar to (17), the above inequality can be written as

$$\left(1 - \widetilde{\eta}_1^2\right) \|g_{k+1}\|^2 \geq \left\| Z_k^T g_{k+1} \right\|^2. \tag{19}$$

It is a challenging task to solve the special subproblem (18). In [16], Hager and Zhang used the L-BFGS method [18,19] to solve the subproblem (18), which causes that the assumptions (9) are imposed on the preconditioners in the convergence analysis of the limited memory CG algorithm. It seems, however, that it is not easy to verify the assumptions (9) in practice. Since the dimension of the subspace $\mathcal{S}_k$ is often small, quasi-Newton method might be a good choice.

For general unconstrained optimization (1), the search direction in quasi-Newton method is the form of $d_k = -B_k^{-1} g_k$, where B_k is a symmetric and positive definite approximation to the Hessian matrix.

Biglari et al. [21] developed a modified BFGS method, in which B_k satisfies the modified secant equation

$$B_{k+1}s_k = y_k^*, \tag{20}$$

where

$$y_k^* = \left(1 + t_k \frac{2\left(f_k - f_{k+1}\right) + \left(g_k + g_{k+1}\right)^T s_k}{s_k^T y_k}\right) y_k \triangleq \gamma_k y_k \tag{21}$$

and $t_k = 2$. It is noted that if $t_k = 0$, then (20) corresponds to the standard secant equation, and if $t_k = 1$, then (20) corresponds to the modified secant equation proposed by Wei et al. [22].

These above secant equations are usually superior to the standard secant equation in the sense that the resulting Hessian approximation contains more accurate curvature information.

Li and Fukushima [23] presented a cautious BFGS method for nonconvex unconstrained optimization, in which B_{k+1} is given by

$$B_{k+1} = \begin{cases} B_k - \frac{B_k s_k s_k^T B_k}{s_k^T B_k s_k} + \frac{y_k y_k^T}{s_k^T y_k}, & \text{if } \frac{s_k^T y_k}{\|s_k\|^2} > \upsilon \|g_k\|^\alpha, \\ B_k, & \text{otherwise,} \end{cases}$$

where $\upsilon > 0$ and $\alpha > 0$.

Motivated by the above quasi-Newton methods, we develop a new quasi-Newton method for solving the subproblem (18).

In what follows, the symbol with hat means that it belongs to the subspace $\mathcal{S}_k$, distinguishing from the symbols in the full space $\mathbb{R}^n$.

Let $\hat{x} = \left(\hat{x}_1, \hat{x}_2, \ldots, \hat{x}_m\right)^T \in \mathbb{R}^m$. The subproblem (18) can be written as

$$\min_{\hat{x} \in \mathbb{R}^m} \hat{f}\left(\hat{x}\right) = f\left(x_k + \hat{x}_1 d_{k-1} + \hat{x}_2 d_{k-2} + \cdots + \hat{x}_m d_{k-m}\right). \tag{22}$$

It follows from the QR decomposition of S_k that $\hat{g}_k = Z_k^T g_k$ and $\hat{y}_k = Z_k^T y_k$. Since the quasi-Newton direction in the subspace is always transformed to the full space, we can easily obtain $d_k = Z_k \hat{d}_k$, $\hat{s}_k^T \hat{y}_k = s_k^T y_k$, $\hat{g}_k^T \hat{s}_k = g_k^T s_k$, $\hat{g}_{k+1}^T \hat{s}_k = g_{k+1}^T s_k$, $\left\|\hat{s}_k\right\|^2 = \|s_k\|^2$ and $\hat{f}_k = f_k$.

According to [24,25],

$$\hat{\mu}_k = \left| \frac{2\left(\hat{f}_{k-1} - \hat{f}_k + \hat{g}_k^T \hat{s}_{k-1}\right)}{\hat{s}_{k-1}^T \hat{y}_{k-1}} - 1 \right| \tag{23}$$

is a quantity showing how $\hat{f}$ is close to a quadratic on the line segment between $\hat{x}_{k-1}$ and $\hat{x}_k$. If the following condition [26,27] holds, namely,

$$\hat{\mu}_k \leq \tau_1 \quad \text{or} \quad \max\left\{\hat{\mu}_k, \hat{\mu}_{k-1}\right\} \leq \tau_2, \tag{24}$$

where τ_1 and τ_2 are small positives and $\tau_1 < \tau_2$, $\hat{f}$ might be very close to a quadratic function on the line segment between $\hat{x}_{k-1}$ and $\hat{x}_k$.

Now we consider the choice of t_k in the subspace version of the modified secant equation (20). When the condition (24) holds, it is natural to use the standard secant equation based on the above observation, which indicates that $t_k = 0$. According to [28], if $\|\hat{s}_k\| > 1$, the Wei's secant equation is expected to be more accurate than the Biglari's secant equation, while if $\|\hat{s}_k\| \leq 1$, the Biglari's secant equation performs better than the Wei's secant equation. Therefore, if $\|\hat{s}_k\|^2 > 1.5$, we choose the Wei's secant equation. Otherwise, we choose the Biglari's secant equation. As a result, we determined t_k as

$$t_k = \begin{cases} 0, & \text{if (24) holds,} \\ 1, & \text{if (24) does not hold and } \left\|\hat{s}_k\right\|^2 > 1.5, \\ 2, & \text{if (24) does not hold and } \left\|\hat{s}_k\right\|^2 \leq 1.5 \end{cases} \tag{25}$$

in the subspace version of the modified secant equation (20).

Motivated by the cautious BFGS method proposed by Li and Fukushima [23], we will set $\hat{B}_k$ to the unit matrix $\hat{I} \in \mathbb{R}^{m\times m}$ when $\frac{\hat{s}_k^T \hat{y}_k}{\hat{s}_k^T \hat{s}_k} \geq \upsilon$, where $\upsilon \geq 10^{-8}$. In addition, $\hat{B}_k$ will be set to $\hat{I}$ after it is updated l times, where

$$l = \max\left(m^2, 45\right). \tag{26}$$

Therefore, the search direction of the new quasi-Newton method for solving the subproblem (22) can be described as

$$\hat{d}_{k+1} = -\hat{B}_{k+1}^{-1} \hat{g}_{k+1}, \tag{27}$$

where $\hat{B}_{k+1}$ is given by

$$\hat{B}_{k+1} = \begin{cases} \hat{B}_k - \frac{\hat{B}_k \hat{s}_k \hat{s}_k^T \hat{B}_k}{\hat{s}_k^T \hat{B}_k \hat{s}_k} + \frac{\hat{y}_k^* \hat{y}_k^{*T}}{\hat{s}_k^T \hat{y}_k^*}, & \text{if } \bmod(k,l) \neq 0 \text{ and } \frac{\hat{s}_k^T \hat{y}_k}{\hat{s}_k^T \hat{s}_k} \geq \upsilon, \\ \hat{I}, & \text{otherwise,} \end{cases} \tag{28}$$

where $\bmod(k,l)$ denotes the residue for k modulo l, $\hat{y}_k^*$ is given by

$$\begin{aligned} \hat{y}_k^* &= \left(1 + t_k \frac{2\left(\hat{f}_k - \hat{f}_{k+1}\right) + \left(\hat{g}_{k+1}^T \hat{s}_k + \hat{g}_k^T \hat{s}_k\right)}{\hat{s}_k^T \hat{y}_k}\right) \hat{y}_k \\ &= \left(1 + t_k \frac{2\left(f_k - f_{k+1}\right) + \left(g_{k+1}^T s_k + g_k^T s_k\right)}{s_k^T y_k}\right) \hat{y}_k \triangleq \gamma_k \hat{y}_k \end{aligned} \tag{29}$$

and t_k is determined by (25). In order to improve the numerical performance and analyze the convergence, we restrict γ_k in (29) as

$$\gamma_k = \min\{\max\{r_1, \gamma_k\}, r_2\}, \tag{30}$$

where $10^{-6} \leq r_1 \leq r_2 \leq 10^8$. Clearly, $\hat{s}_k^T \hat{y}_k^* > 0$, and thus it is not difficult to verify that $\hat{B}_{k+1}$ is symmetric and positive definite when $\hat{B}_k$ is symmetric and positive definite and $\hat{s}_k^T \hat{y}_k^* > 0$.

In our algorithm, the search direction (27) in the subspace will be transformed to the full space $\mathbb{R}^n$ at each subspace iteration, namely,

$$d_{k+1} = -P_k g_{k+1}, \tag{31}$$

where

$$P_k = Z_k \hat{B}_{k+1}^{-1} Z_k^T, \tag{32}$$

and $\hat{B}_{k+1}$ is given by (28).

Obviously, if $\hat{B}_{k+1} \neq \hat{I}$, then it follows from (28) and (29) that $\gamma_k \hat{B}_{k+1}^{-1} \hat{y}_k = \hat{s}_k$, and thus P_k in (32) satisfies

$$P_k y_k = \frac{1}{\gamma_k} s_k. \tag{33}$$

For β_k^{PDK} in (11) and β_k^{PDK+} in (13), we have the following result.

Lemma 2.1 *Suppose that P_k be a symmetric and positive definite quasi-Newton approximation to the inverse of Hessian matrix which satisfies the modified secant equation* (33)*. Then,*

$$\beta_k^{PDK} = \beta_k^{PDK+} = 0.$$

Proof It follows from the secant condition (33) that

$$\begin{aligned}\beta_k^{PDK} &= \frac{g_{k+1}^T P_k y_k}{d_k^T y_k} - \frac{y_k^T P_k y_k}{d_k^T y_k} \frac{g_{k+1}^T d_k}{d_k^T y_k} = \frac{g_{k+1}^T s_k}{\gamma_k d_k^T y_k} - \frac{y_k^T s_k}{\gamma_k d_k^T y_k} \frac{g_{k+1}^T d_k}{d_k^T y_k} \\ &= \frac{g_{k+1}^T s_k}{\gamma_k d_k^T y_k} - \frac{g_{k+1}^T s_k}{\gamma_k d_k^T y_k} = 0.\end{aligned}$$

According to (13), we know that if $\beta_k^{PDK} = 0$, then $\beta_k^{PDK+} = \beta_k^{PDK}$, which implies that $\beta_k^{PDK+} = \beta_k^{PDK} = 0$. The proof is completed. □

It follows from Lemma 2.1 that the search direction (31) with $\hat{B}_{k+1} \neq \hat{I}$ can be regarded as the preconditioned CG direction (12) with the preconditioner P_k in (32) with $\hat{B}_{k+1} \neq \hat{I}$.

Once the orthogonality is improved, the iteration will depart the subspace to enter the full space, and the standard CG iteration is invoked immediately.

2.2 A modified strategy for the choice of the initial stepsize

It is universally acknowledged that the choice of the initial stepsize is crucial to an optimization method. Unlike general quasi-Newton methods, it is challenging to determine a suitable initial stepsize for CG algorithms. In the subsection, we design a modified strategy for choosing the initial stepsize based on the strategy in [8].

Denote

$$\phi_k(\alpha) = f(x_k + \alpha d_k), \alpha \geq 0.$$

Hager and Zhang [9] chose the initial stepsize in CG_DESCENT as follows:

$$\alpha_k^0 = \begin{cases} \arg\min\ q\left(\phi_k(0), \phi_k'(0), \phi_k(\bar{\tau}_1\alpha_{k-1})\right), & \text{if } \phi_k(\bar{\tau}_1\alpha_{k-1}) \leq \phi_k(0), \\ \bar{\tau}_2\alpha_{k-1}, & \text{otherwise,} \end{cases} \tag{34}$$

where $\bar{\tau}_1 > 0$, $\bar{\tau}_2 > 0$ and $q\left(\phi_k(0), \phi_k'(0), \phi_k(\bar{\tau}_1\alpha_{k-1})\right)$ is a interpolation function matched the three values $\phi_k(0)$, $\phi_k'(0)$ and $\phi_k(\bar{\tau}_1\alpha_{k-1})$. Dai and Kou [8] determined the initial stepsize in CGOPT as follows:

$$\alpha_k^0 = \begin{cases} \alpha, & \text{if } |\phi_k(\alpha) - \phi_k(0)| / (\tau_3 + |\phi_k(0)|) > \tau_4, \\ \arg\min\ q\left(\phi_k(0), \phi_k'(0), \phi_k(\alpha)\right), & \text{otherwise,} \end{cases} \tag{35}$$

where

$$\alpha = \max\left\{\tau_5\alpha_{k-1}, -2|f_k - f_{k-1}| / g_k^T d_k\right\}, \tag{36}$$

$\tau_3 > 0$, $\tau_4 > 0$ and $\tau_5 > 0$. Recently, motivated by the BB method [29] and using the interpolation technique, Liu and Liu [11] also developed an efficient strategy for the choice of the initial stepsize for the subspace minimization conjugate gradient method (SMCG_BB).[1]

If the search direction d_k is determined by (31) with $\hat{B}_{k+1} \neq \hat{I}$, then the trial initial stepsize $\bar{\alpha}$ should be taken as 1 like quasi-Newton methods. Otherwise, the trial initial stepsize is determined by (36). As a result, the trial initial stepsize is described as

$$\bar{\alpha} = \begin{cases} 1, & \text{if } d_k \text{ is computed by (31) with } \hat{B}_{k+1} \neq \hat{I}, \\ \alpha, & \text{otherwise,} \end{cases} \tag{37}$$

where α is given by (36).

It is well-known that the linear CG algorithm with the exact line search enjoys quadratic termination for strictly convex quadratic functions. In addition, Andrei [30]

[1] Available at https://web.xidian.edu.cn/xdliuhongwei/en/paper.html.

thought that the higher accuracy of the stepsize, the faster convergence of a CG algorithm. If the quantity $\mu_k = \left| \frac{2(f_{k-1}-f_k+g_k^T s_{k-1})}{s_{k-1}^T y_{k-1}} - 1 \right|$ [24,25] satisfies

$$\mu_k \leq \tau \quad \text{or} \quad \max\{\mu_k, \mu_{k-1}\} \leq \bar{\tau}, \tag{38}$$

where

$$\tau = \begin{cases} \tau_6, & \text{if } d_k \text{ is computed by (31) with } \hat{B}_{k+1} \neq \hat{I}, \\ \tau_7, & \text{otherwise,} \end{cases}$$

$$\bar{\tau} = \begin{cases} \tau_8, & \text{if } d_k \text{ is computed by (31) with } \hat{B}_{k+1} \neq \hat{I}, \\ \tau_9, & \text{otherwise,} \end{cases}$$

then f might be close to a quadratic function on the section between x_{k-1} and x_k [26,27]. Here $0 < \tau_6 < \tau_8$ and $0 < \tau_7 < \tau_9$. It is easy to verify that $\hat{\mu}_k$ in (23) is the subspace version of the above quantity μ_k. Based on the above observations, when f is close to a quadratic function on the section between x_{k-1} and x_k, it is also reasonable to take the minimizer of the interpolation function $q\left(\phi_k(0), \phi_k'(0), \phi_k(\bar{\alpha})\right)$ as the initial stepsize, where $\bar{\alpha}$ is given by (37).

Therefore, the initial stepsize is determined by

$$\alpha_k^0 = \begin{cases} \arg\min\ q\left(\phi_k(0), \phi'_k(0), \phi_k(\bar{\alpha})\right), \\ \bar{\alpha}, \end{cases} \quad \begin{array}{l} \text{if (38) or } \frac{|\phi_k(\bar{\alpha})-\phi_k(0)|}{(\tau_3+|\phi_k(0)|)} > \widehat{\tau} \text{ holds,} \\ \text{otherwise,} \end{array} \tag{39}$$

where the trial stepsize $\bar{\alpha}$ is determined by (37), τ_3 is the same as that in (35) and

$$\widehat{\tau} = \begin{cases} \tau_{10}, & \text{if } d_k \text{ is computed by (31) with } \hat{B}_{k+1} \neq \hat{I}, \\ \tau_{11}, & \text{otherwise.} \end{cases}$$

where $\tau_{10} > 0$ and τ_{11} are positive parameters.

2.3 Description of the improved Dai–Kou CG algorithm

Based on [8], we describe the improved Dai–Kou CG algorithm for unconstrained optimization in detail. As mentioned above, the improved CG algorithm consists of two kinds of iterations. The term "status" in Algorithm 1 stands for the type of iteration, namely, status= "standard CG" indicates the standard CG iteration will be performed, and status= "subspace" indicates the subspace iteration will be performed.

Remark 1 It is worth noting that when the orthogonality is improved, the iteration will depart the subspace and the standard CG iteration is invoked immediately. Whereas the limited memory CG algorithm [16] first performs the special preconditioned CG iteration corresponding to the third preconditioner in (8) when departing the subspace. It indicates that Algorithm 1 is simpler than the limited memory CG algorithm [16].

Algorithm 1 (The improved Dai–Kou CG algorithm)

Step 0 Initialization. Given $x_0 \in \mathbb{R}^n$, $\varepsilon > 0$, $\epsilon_4 > 0$, η, $\widetilde{\eta}_0$, $\widetilde{\eta}_1$, τ_1, τ_2, τ_6, τ_7, τ_8, τ_9, τ_{10}, τ_{11}, υ, r_1, r_2, MaxRestart, MinQuad, IterQuad=0 and IterRestart=0. Set status= "standard CG" and $k = 0$.

Step 1 If $||g_k||_\infty \leq \varepsilon$, then stop.

Step 2 Compute the search direction.

If (status ="standard CG"), then

If $k = 0$, then $d_0 = -g_0$.

elseif (IterRestart=MaxRestart or (IterQuad=MinQuad and IterQuad ≠IterRestart)), then

$d_k = -g_k$ and set IterRestart = 0, IterQuad = 0.

else

set $P_k = I$, compute d_k by (15).

end

elseif (status ="subspace"), then

Compute P_k by (32), and determine the search direction d_k by (31).

end

Step 3 Determine a stepsize α_k satisfying (5) and (6) with the initial stepsize (39).

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

Step 5 Update IterRestart and IterQuad. IterRestart = IterRestart + 1. If $\left|\frac{2(f_{k+1}-f_k)}{(g_{k+1}+g_k)^T s_k} - 1\right| \leq \epsilon_4$ or $\left|f_{k+1} - f_k - 0.5\left(g_{k+1}^T s_k + g_k^T s_k\right)\right| \leq \epsilon_4$ [11], then IterQuad = IterQuad +1; otherwise, IterQuad = 0.

Step 6 Update the type of iteration.

If (status ="standard CG"), then

If the condition (17) holds, then status ="subspace".

elseif (status ="subspace"), then

If the condition (19) holds, then status ="standard CG".

end

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

We describe some implementation details about Algorithm 1 here. The Gram–Schmidt orthogonality method [17] is used to calculate the QR factorization: $S_k = Z_k \bar{R}_k$, and thus Z_k is computed by $Z_k = S_k \bar{R}_k^{-1}$. For the search direction (31), we first calculate the modified Cholesky factorization of $\hat{B}_k$: $\hat{B}_k = \hat{L}_k \hat{D}_k \hat{L}_k^T$, where $\hat{L}_k \in \mathbb{R}^{m\times m}$ is a unit lower triangular matrix and $\hat{D}_k \in \mathbb{R}^{m\times m}$ is a diagonal matrix, and determine the search search in the subspace by $\hat{L}_k \hat{D}_k \hat{L}_k^T \hat{d}_k = -\hat{g}_k$ and then compute the search direction (31) by $d_k = Z_k \hat{d}_k = S_k \bar{R}_k^{-1} \hat{d}_k$.

3 Convergence analysis

In the section, we study some important properties of $P_k = I$ and P_k in (32) and establish the global convergence of Algorithm 1 under the following assumptions.

Assumption 3.1 (i)The objective function f is continuously differentiable on $\mathbb{R}^n$; (ii) The level set $\mathcal{L} = \left\{ x | f(x) \leq f(x_0) + \sum_{k\geq 0} \overline{\eta}_k \right\}$ is bounded; (iii) The gradient g is Lipschitz continuous, namely, there exists a constant $L > 0$ such that

$$\| g(x) - g(y) \| \leq L \| x - y \|, \quad \forall x, y \in \mathbb{R}^n.$$

Lemma 3.1 *Suppose Assumption 3.1 hold. Then, for $\hat{B}_{k+1}$ in (28), there exist three constants $\hat{\xi}_1 > 0, \hat{\xi}_2 > 0$ and $\hat{\xi}_3 > 0$ such that*

$$\lambda_{\max}\left(\hat{B}_k\right) \leq \hat{\xi}_1, \qquad \lambda_{\max}\left(\hat{B}_k^{-1}\right) \leq \hat{\xi}_2, \qquad \left\|\hat{B}_k^{-1}\right\| \leq \hat{\xi}_3.$$

Proof Since the columns of Z_k forms the normal orthogonal basis for $\mathcal{S}_k$ and $m < +\infty$, there exists $\xi_0 > 0$ such that $\|Z_k\| \leq \xi_0$. According to (28), (30) and the equivalence of matrix norm in the finite dimensional space, we have that

$$\lambda_{\max}\left(\hat{B}_{k+1}\right) = 1, \tag{40}$$

or

$$\begin{aligned}
\lambda_{\max}\left(\hat{B}_{k+1}\right) &\leq \lambda_{\max}\left(\hat{B}_k\right) + \lambda_{\max}\left(-\frac{\hat{B}_k \hat{s}_k \hat{s}_k^T \hat{B}_k}{\hat{s}_k^T \hat{B}_k \hat{s}_k}\right) + \lambda_{\max}\left(\frac{\hat{y}_k^* \hat{y}_k^{*T}}{\hat{s}_k^T \hat{y}_k^*}\right) \\
&\leq \lambda_{\max}\left(\hat{B}_k\right) + \gamma_k \frac{\hat{y}_k^T \hat{y}_k}{\hat{s}_k^T \hat{y}_k} \\
&\leq \hat{\lambda}_{\max}\left(\hat{B}_k\right) + \gamma_k L^2 \xi_0^2 \frac{\left\|\hat{s}_k\right\|^2}{\hat{s}_k^T \hat{y}_k} \\
&\leq \hat{\lambda}_{\max}\left(\hat{B}_k\right) + \frac{r_2}{\upsilon} L^2 \xi_0^2.
\end{aligned}$$

The above first inequality is obtained by the property of $\lambda_{\max}(\cdot)$: $\lambda_{\max}(A_1 + A_2) \leq \lambda_{\max}(A_1) + \lambda_{\max}(A_2)$, where $A_1 \in \mathbb{R}^{m\times m}$ and $A_2 \in \mathbb{R}^{m\times m}$ are symmetric matrices, and the third comes from $\hat{y}_k = Z_k^T y_k$, $\|Z_k\| \leq \xi_0$ and Assumption 3.1 (iii). Since $\hat{B}_k$ will be set to $\hat{I}$ after updating at most l times, where l is given by (26). Therefore, we obtain that $\lambda_{\max}\left(\hat{B}_{k+1}\right) \leq 1 + \frac{l r_2 L^2 \xi_0^2}{\upsilon} \triangleq \hat{\xi}_1$.

Let $\hat{P}_k = \hat{B}_{k+1}^{-1}$. According to (28), after some simple matrix operations we obtain that

$$\hat{P}_k = \hat{I}$$

or

$$\hat{P}_k = \left(\hat{I} - \frac{\hat{y}_k \hat{s}_k^T}{\hat{s}_k^T \hat{y}_k}\right)^T \hat{P}_{k-1} \left(\hat{I} - \frac{\hat{y}_k \hat{s}_k^T}{\hat{s}_k^T \hat{y}_k}\right) + \frac{1}{\gamma_k} \frac{\hat{s}_k \hat{s}_k^T}{\hat{s}_k^T \hat{y}_k}. \tag{41}$$

It is not difficult to see that $\lambda_{\max}\left(\left(\hat{I}-\frac{\hat{y}_k\hat{s}_k^T}{\hat{s}_k^T\hat{y}_k}\right)^T\left(\hat{I}-\frac{\hat{y}_k\hat{s}_k^T}{\hat{s}_k^T\hat{y}_k}\right)\right)=\frac{\|\hat{y}_k\|^2\|\hat{s}_k\|^2}{(\hat{s}_k^T\hat{y}_k)^2}$. For any $\hat{z}\neq 0\in\mathbb{R}^m$ and $\hat{P}_k$ in (41), we have from the property of $\lambda_{\max}(\cdot)$ and Cauchy inequality that

$$\begin{aligned}\hat{z}^T\hat{P}_k\hat{z}&=\hat{z}^T\left(\hat{I}-\frac{\hat{y}_k\hat{s}_k^T}{\hat{s}_k^T\hat{y}_k}\right)^T\hat{P}_{k-1}\left(\hat{I}-\frac{\hat{y}_k\hat{s}_k^T}{\hat{s}_k^T\hat{y}_k}\right)\hat{z}+\frac{1}{\gamma_k}\frac{(\hat{s}_k^T\hat{z})^2}{\hat{s}_k^T\hat{y}_k}\\&\le\hat{\lambda}_{\max}\left(\hat{P}_{k-1}\right)\hat{z}^T\left(\hat{I}-\frac{\hat{y}_k\hat{s}_k^T}{\hat{s}_k^T\hat{y}_k}\right)^T\left(\hat{I}-\frac{\hat{y}_k\hat{s}_k^T}{\hat{s}_k^T\hat{y}_k}\right)\hat{z}+\frac{1}{\gamma_k}\frac{(\hat{s}_k^T\hat{z})^2}{\hat{s}_k^T\hat{y}_k}\\&\le\lambda_{\max}\left(\hat{P}_{k-1}\right)\lambda_{\max}\left(\left(\hat{I}-\frac{\hat{y}_k\hat{s}_k^T}{\hat{s}_k^T\hat{y}_k}\right)^T\left(I-\frac{\hat{y}_k\hat{s}_k^T}{\hat{s}_k^T\hat{y}_k}\right)\right)\|\hat{z}\|^2+\frac{1}{\gamma_k}\frac{(\hat{s}_k^T\hat{z})^2}{\hat{s}_k^T\hat{y}_k}\\&\le\hat{\lambda}_{\max}\left(\hat{P}_{k-1}\right)\frac{\|\hat{y}_k\|^2\|\hat{s}_k\|^2}{(\hat{s}_k^T\hat{y}_k)^2}\|\hat{z}\|^2+\frac{1}{\gamma_k}\frac{\|\hat{s}_k\|^2}{\hat{s}_k^T\hat{y}_k}\|\hat{z}\|^2.\end{aligned}$$

Dividing the above inequality by $\|\hat{z}\|^2$ and maximizing the resulting inequality, we can obtain from (28) that

$$\begin{aligned}\lambda_{\max}\left(\hat{P}_k\right)&\le\lambda_{\max}\left(\hat{P}_{k-1}\right)\frac{\|\hat{y}_k\|^2\|\hat{s}_k\|^2}{(\hat{s}_k^T\hat{y}_k)^2}+\frac{1}{\gamma_k}\frac{\|\hat{s}_k\|^2}{\hat{s}_k^T\hat{y}_k}\\&\le\lambda_{\max}\left(\hat{P}_{k-1}\right)L^2\xi_0^2\frac{\|\hat{s}_k\|^4}{(\hat{s}_k^T\hat{y}_k)^2}+\frac{1}{\gamma_k}\frac{\|\hat{s}_k\|^2}{\hat{s}_k^T\hat{y}_k}\\&\le\frac{L^2\xi_0^2}{v^2}\lambda_{\max}\left(\hat{P}_{k-1}\right)+\frac{1}{r_1 v}.\end{aligned}$$

The above second inequality comes from $\hat{y}_k=Z_k^T y_k$, $\|Z_k\|\le\xi_0$ and Assumption 3.1 (iii). Since $\hat{P}_k$ will be set to $\hat{I}$ after updating at most l times, we know easily there exists a $\hat{\xi}_2>0$ such that $\lambda_{\max}\left(\hat{B}_{k+1}^{-1}\right)=\lambda_{\max}\left(\hat{P}_k\right)\le\hat{\xi}_2$.

Since $\hat{B}_{k+1}^{-1}$ is a symmetric and positive definite matrix, we have that $\left\|\hat{B}_{k+1}^{-1}\right\|_2^2=\lambda_{\max}\left(\hat{B}_{k+1}^{-1}\right)\le\hat{\xi}_2$. Therefore, by the equivalence of matrix norm in finite dimensional space, we know that there exists a constant $\hat{\xi}_3>0$ such that $\left\|\hat{B}_{k+1}^{-1}\right\|<\hat{\xi}_3$. The proof is completed. □

Lemma 3.2 *Suppose Assumption 3.1 hold. Then, for $P_k=I$ or P_k in (32), there exist three constants $\gamma_0>0$, $\gamma_1>0$ and $\gamma_2>0$ such that*

$$\|P_k\|\le\gamma_0,\quad g_{k+1}^T P_k g_{k+1}\ge\gamma_1\|g_{k+1}\|^2,\quad d_k^T P_k^{-1}d_k\ge\gamma_2\|d_k\|^2.$$

Proof We consider the following two cases.

(i) $P_k = I$. It is obvious that the conclusions hold.

(ii) $P_k = Z_k \hat{B}_{k+1}^{-1} Z_k^T$. By (19), (32), Lemma 3.1 and Assumption 3.1 (iii), we obtain that

$$\begin{aligned}
\|P_k\| &= \left\| Z_k \hat{B}_{k+1}^{-1} Z_k^T \right\| = \left\| \hat{B}_{k+1}^{-1} \right\| \leq \xi_3 \triangleq \gamma_0, \\
g_{k+1}^T P_k g_{k+1} &= g_{k+1}^T Z_k \hat{B}_{k+1}^{-1} Z_k^T g_{k+1} = \hat{g}_{k+1}^T \hat{B}_{k+1}^{-1} \hat{g}_{k+1} \geq \lambda_{\min}\left(\hat{B}_{k+1}^{-1}\right) \left\|\hat{g}_{k+1}\right\|^2 \\
&\geq \frac{1}{\hat{\xi}_1}\left(1 - \widetilde{\eta}_1^2\right) \|g_{k+1}\|^2 \triangleq \gamma_1 \|g_{k+1}\|^2, \\
d_k^T P_k^{-1} d_k &= d_k^T Z_k \hat{B}_{k+1} Z_k^T d_k = \hat{d}_k^T \hat{B}_{k+1} \hat{d}_k \geq \frac{1}{\hat{\xi}_2} \left\|\hat{d}_k\right\|^2 = \frac{1}{\hat{\xi}_2} \|d_k\|^2 \triangleq \gamma_2 \|d_k\|^2.
\end{aligned}$$

The proof is completed. □

Remark 2 By Lemmas 3.1 and 3.2 , we know that $P_k = I$ or P_k in (32) in Algorithm 1 satisfies the conditions (9), while the preconditioners (8) in the limited memory CG methods [16] are artificially assumed to satisfy the conditions (9).

Lemma 3.3 *Assume that f satisfies Assumption 3.1, and let $\{x_k\}$ be the sequence generated by Algorithm 1. If $d_k^T y_k \neq 0$, then there exists a constant $c > 0$ such that*

$$g_{k+1}^T d_{k+1} \leq -c\|g_{k+1}\|^2. \tag{42}$$

Proof We consider the following two cases.

Case I. Standard CG iteration

(i) If $\beta_k^{PDK+} = \beta_k^{DK}$, then it follows from Lemma 2.2 in [8] that

$$g_{k+1}^T d_{k+1} \leq -\frac{3}{4}\|g_{k+1}\|^2. \tag{43}$$

(ii)If $\beta_k^{PDK+} = \eta_k$, where η_k is given by (14), then

$$\begin{aligned}
g_{k+1}^T d_{k+1} &= -\|g_{k+1}\|^2 - \eta \frac{\left|g_{k+1}^T d_k\right|}{\|d_k\|^2} g_{k+1}^T d_k \\
&\leq -\|g_{k+1}\|^2 + \eta \|g_{k+1}\|^2 = -(1-\eta)\|g_{k+1}\|^2.
\end{aligned} \tag{44}$$

Case II. Subspace iteration.

According to Lemma 3.2, we have that

$$g_{k+1}^T d_{k+1} = -g_{k+1}^T P_k g_{k+1} \leq -\gamma_1 \|g_{k+1}\|^2. \tag{45}$$

In sum, we know from (43), (44) and (45) that the search direction d_{k+1} satisfies the sufficient descent condition (42) with $c = \min\{3/4, 1-\eta, \gamma_1\}$. The proof is completed. □

In the following analysis, the search direction (31) is treated as $d_{k+1} = -P_k g_{k+1} + \beta_k^{PDK+} d_k$ with $\beta_k^{PDK+} = \max\left\{0, -\eta \frac{|g_{k+1}^T d_k|}{d_k^T P_k^{-1} d_k}\right\}$. The next lemma is used to establish the convergence of Algorithm 1.

Lemma 3.4 *Assume that f satisfies Assumption 3.1, and let $\{x_k\}$ be generated by Algorithm 1. If $\gamma = \inf\{\|g_k\| : k \geq 1\} > 0$, then $d_k \neq 0$ and $\sum_{k\geq 0} \|u_k - u_{k-1}\|^2 < +\infty$, where $u_k = d_k / \|d_k\|$.*

Proof By $\gamma > 0$ and Lemma 3.3 we know that $\|d_k\| \neq 0$. By (5), (6) and Lemma 3.3, we get that

$$\sum_{k=0}^{+\infty} \frac{1}{\|d_k\|^2} < +\infty. \tag{46}$$

Similar to Lemma 4.3 of [8], by $\|P_k\| \leq \gamma_0$ and $d_k^T P_k^{-1} d_k \geq \gamma_2 \|d_k\|^2$ in Lemma 3.2, we can obtain that $\sum_{k\geq 0} \|u_k - u_{k-1}\|^2 < +\infty$. □

The convergence of Algorithm 1 is established in the following theorem.

Theorem 3.1 *Assume f satisfies Assumption 3.1, let $\{x_k\}$ be the sequence generated by Algorithm 1. Then,*

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

Proof Clearly, $\{x_k\} \subset \mathcal{L}$. It follows from Assumption 3.1 (i) and the boundedness of $\mathcal{L}$ that $\Gamma = \max_{x\in\mathcal{L}} \|g(x)\| < +\infty$. We prove the conclusion by contradiction. We suppose that $\lim_{k\to\infty} \inf \|g_k\| > 0$ and $g_k \neq 0$ for all k. Therefore, we have that $\|g_k\| \neq 0$ and $\gamma = \inf\{\|g_k\| : k \geq 0\} > 0$. The proof is divided into the following three steps:

I. A bound for β_k^{PDK+}. We consider the following two cases. (i) The search direction d_k is computed by (15). If $\beta_k^{DK} \geq 0$, then $\beta_k^{PDK+} = \beta_k^{DK}$, otherwise $\beta_k^{PDK+} \geq \beta_k^{DK}$, which implies that $\left|\beta_k^{PDK+}\right| \leq \left|\beta_k^{DK}\right|$. According to (6), Assumption 3.1 (iii) and Lemma 3.3, similar to Step I of Theorem 3.2 in [9] we can obtain that $\left|\beta_k^{PDK+}\right| \leq \left|\beta_k^{DK}\right| \leq C \|s_k\|$, where

$$C = \frac{1}{c(1-\sigma)\gamma^2}\left(L\Gamma + L^2 D \max\left\{\frac{\sigma}{1-\sigma}, 1\right\}\right), D = \max\{\|y - z\|, \forall y, z \in \mathcal{L}\}. \tag{47}$$

(ii) The search direction d_k is computed by (31). Obviously, $\left|\beta_k^{PDK+}\right| = 0 \leq C \|s_k\|$, where C is given by (47).

II. A bound on the steps $\|s_k\|$. According to Lemma 3.4, similar to Step II of Theorem 3.2 in [9] we can obtain that $\sum_{j=k}^{l-1} \|s_j\| \leq 2D$ when $l > k \geq k_0$ and $l - k \leq \Delta$, where k_0 is chosen such that $\sum_{i \geq k_0} \|u_{i+1} - u_i\|^2 \leq \frac{1}{4\Delta}$ and Δ is a positive integer satisfying $\Delta \geq 4CD$. Here C and D are given by (47).

III. A bound on d_l. According to (12), the bound for β_k^{PDK+} mentioned above and $\|P_k\| \leq \gamma_0$ in Lemma 3.2, we have that $\|d_l\|^2 \leq \left(\|-P_{l-1}g_l\| + \left|\beta_{l-1}^{PDK+}\right| \|d_{l-1}\|\right)^2 \leq 2\gamma_0^2\Gamma^2 + 2C^2\|s_{l-1}\|^2\|d_{l-1}\|^2$. Similar to Step III of Theorem 3.2 in [9] we know that $\|d_l\|$ is bounded and the bound is independent of $l > k_0$, which contracts with (46). Therefore, we get $\liminf_{k\to\infty} \|g_k\| = 0$. The proof is completed. □

Remark 3 It is worth noting that although a new quasi-Newton method is developed to solve the subspace problem (22), the convergence of Algorithm 1 is established under the quite mild Assumption 3.1 without the strict conditions (9).

4 Numerical experiments

Since the original Dai–Kou CG algorithm [8] corresponds to the version 1.0 of CGOPT, namely, CGOPT (1.0), we refer to the improved Dai–Kou CG algorithm as the version 2.0 of CGOPT, namely, CGOPT (2.0). We do some experiments to compare CGOPT (2.0) with CGOPT (1.0) and the latest limited memory CG software package CG_DESCENT (6.8). CGOPT (2.0) is implemented based on the C code of CGOPT (1.0), and the codes of CGOPT (1.0) and CG_DESCENT (6.8) can be downloaded from http://coa.amss.ac.cn/wordpress/?page_id=21 and http://users.clas.ufl.edu/hager/papers/Software, respectively. The test collection includes 160 unconstrained optimization problems from the CUTEr library [31], which can be found in http://users.clas.ufl.edu/hager/papers/CG/results6.0.txt; the initial points and the dimensions of the test problems are default.

In the numerical experiments, we choose the following parameters for CGOPT (2.0): $\varepsilon = 10^{-6}$, $\eta = 0.3$, $\widetilde{\eta}_0 = 10^{-6}$, $\widetilde{\eta}_1 = 0.4$, $\tau_1 = 10^{-8}$, $\tau_2 = 10^{-4}$, $\tau_6 = 10^{-4}$, $\tau_7 = 5 \times 10^{-3}$, $\tau_8 = 10^{-3}$, $\tau_9 = 5 \times 10^{-2}$, $\tau_{10} = 50$, $\tau_{11} = 110$, $\bar{\eta}_k = 1/(k^{1.4})$, $\upsilon = 5 \times 10^{-6}$, $r_1 = 10^{-4}$, $r_2 = 10^6$ and $m = \min\{11, n\}$, and use other default parameter values in CGOPT (1.0). CG_DESCENT (6.8) and CGOPT (1.0) use all default parameter values in their codes except the stopping conditions, which means that CG_DESCENT (6.8) adaptively uses the standard Wolfe line search or the AWolfe line search at each iteration. All test methods are terminated if $\|g_k\|_\infty \leq 10^{-6}$ is satisfied.

The performance profiles introduced by Dolan and Moré [32] are used to display the performances of these algorithms. In Figs. 1, 2, 3, 4, 5, 6, 7, 8, "N_{iter}", "N_f", "N_g" and "T_{cpu}" represent the number of iterations, the number of function evaluations, the number of gradient evaluations and CPU time (s), respectively.

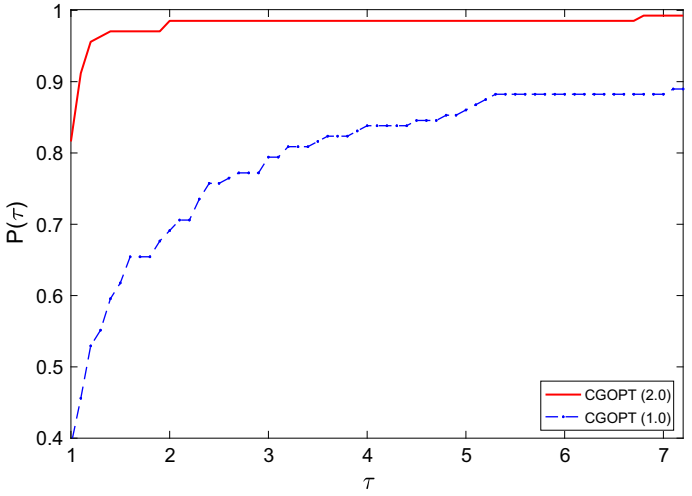

Fig. 1 N_{iter}

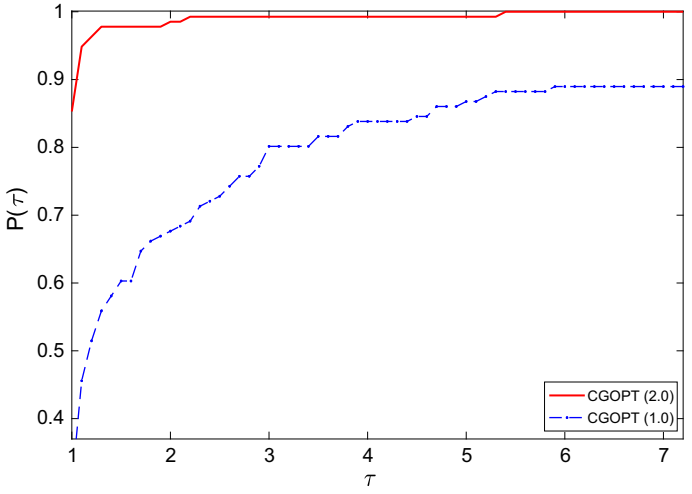

Fig. 2 N_f

In the numerical experiments, CGOPT (2.0) solves successfully 142 problems, while CG_DESCENT (6.8) and CGOPT (1.0) solve successfully 145 and 137 problems, respectively.

Figures 1, 2, 3, and 4 plot the performance profiles of CGOPT (2.0) and CGOPT (1.0) in term of the number of iterations, the number of function evaluations, the number of gradient evaluations and CPU time. As shown in Figs. 1, 2, 3 and 4 we observe that CGOPT (2.0) has a quite significant improvement over CGOPT (1.0) in term of the numbers of iterations, function evaluations and gradient evaluations and CPU time. It indicates that CGOPT (2.0) is superior much to CGOPT (1.0) .

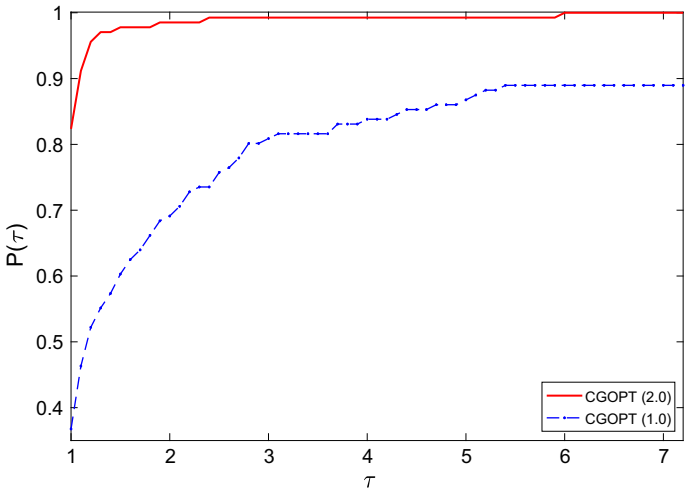

Fig. 3 N_g

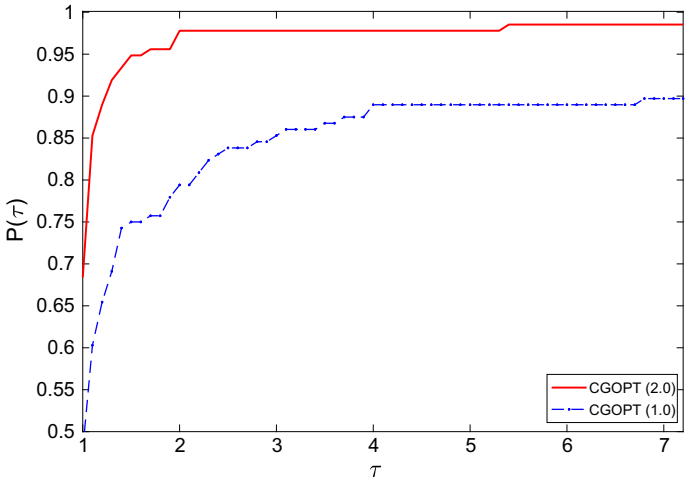

Fig. 4 T_{cpu}

Figures 5, 6, 7 and 8 plot the performance profiles of CGOPT (2.0) and CG_DESCENT (6.8) in term of the number of iterations, the number of function evaluations, the number of gradient evaluations and CPU time. As showed in Fig. 5, we see that CGOPT (2.0) performs better than CG_DESCENT (6.8) in term of the numbers of iterations, since CGOPT (2.0) is better for about 66% of the test problems, while the percentage of CG_DESCENT (6.8) is only about 50%. We observe from Fig. 6 that CGOPT (2.0) performs slightly better than CG_DESCENT (6.8) in term of the number of function evaluations. In Fig. 7, we see that CGOPT (2.0) outperforms much CG_DESCENT (6.8) in term of the number of gradient evaluations, since CGOPT (2.0) is better for about 74% of the test problems, while the

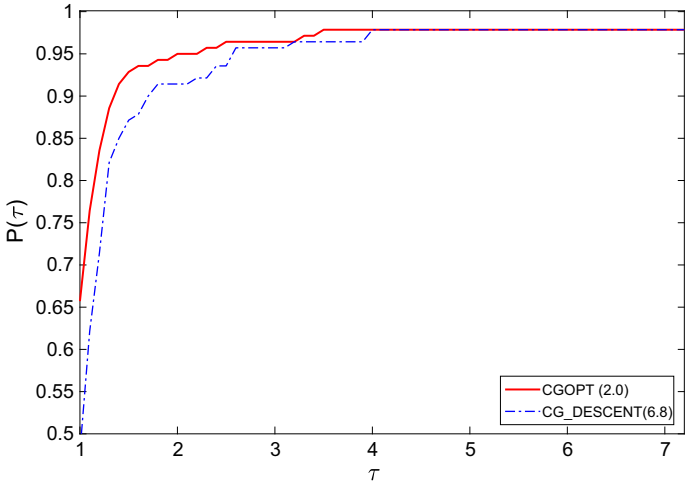

Fig. 5 N_{iter}

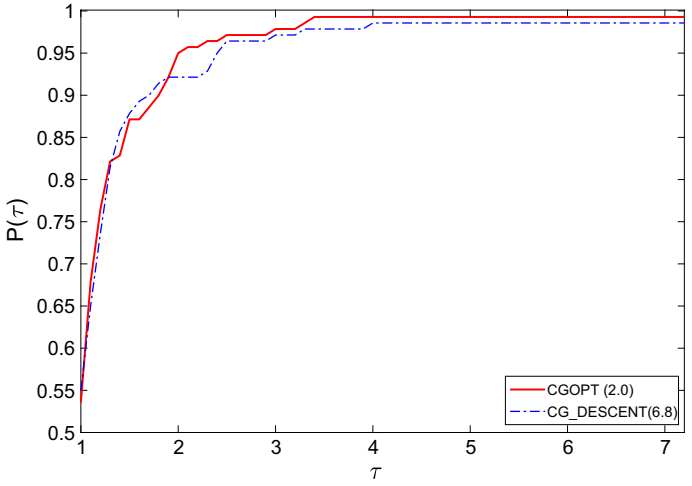

Fig. 6 N_f

percentage of CG_DESCENT (6.8) is only about 36%. We observe from Fig. 8 that CGOPT (2.0) is faster than CG_DESCENT (6.8). It follows from Theorem 3.1 that CGOPT (2.0) with the improved Wolfe line search used is globally convergent, whereas there is no guarantee for the global convergence of CG_DESCENT with the quite efficient AWolfe line search. It indicates that CGOPT (2.0) is superior to CG_DESCENT (6.8) for the CUTEr library in theory and numerical performance.

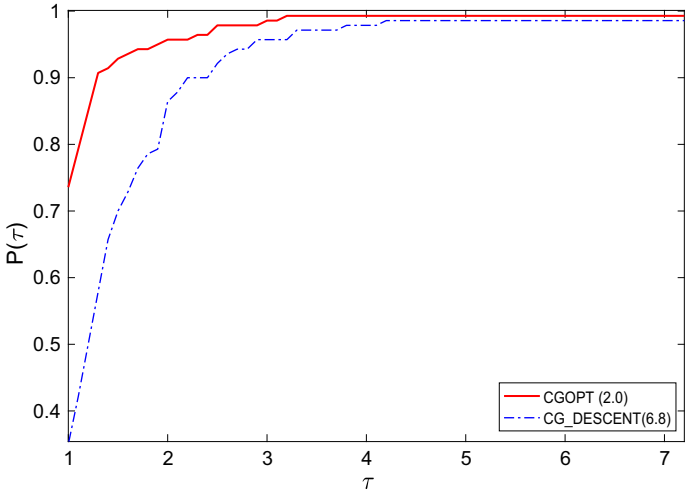

Fig. 7 N_g

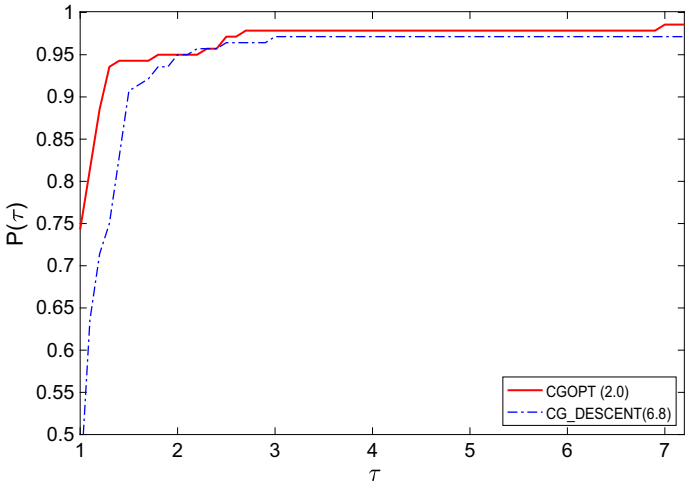

Fig. 8 T_{cpu}

5 Conclusions

To overcome the three drawbacks in the limited memory CG algorithm [16], we present an improved Dai–Kou CG algorithm for unconstrained optimization, which consists of two kinds of iterations. In the improved Dai–Kou CG algorithm, a new quasi-Newton method for improving the orthogonality and a modified strategy for choosing the initial stepsize are analyzed. We establish the convergence of the improved Dai–Kou CG algorithm without the assumptions (9). Some numerical results indicate that the improved Dai–Kou CG algorithm (CGOPT (2.0)) has a great improvement over

the original Dai–Kou CG algorithm (CGOPT (1.0)) and outperforms the latest limited memory CG software package CG_DESCENT (6.8) for the CUTEr library.

Acknowledgements We would like to thank the anonymous referees for their useful comments. We also would like to thank professors Hager, W. W. and Zhang, H. C. for their C code of CG_DESCENT (6.8). The third author's work was partly supported by the Chinese NSF grants (Nos. 11631013 and 11971372) and Key Project of Chinese National Programs for Fundamental Research and Development (No. 2015CB856002). The first author's work was supported by the National Natural Science Foundation of China (no. 11901561) and the Natural Science Foundation of Guangxi(No. 2018GXNSFBA281180).

References

1. Fletcher, R., Reeves, C.: Function minimization by conjugate gradients. Comput. J. **7**(2), 149–154 (1964)
2. Hestenes, M.R., Stiefel, E.L.: Methods of conjugate gradients for solving linear systems. J. Res. Natl. Bur. Stand. **49**(6), 409–436 (1952)
3. Polak, E., Ribière, G.: Note sur la convergence de méthodes de directions conjugées. Rev. Fr. Inform. Rech. Opér. **3**, 35–43 (1969)
4. Polyak, B.T.: The conjugate gradient method in extreme problems. USSR Comput. Math. Math. Phys. **9**, 94–112 (1969)
5. Shanno, D.F.: On the convergence of a new conjugate gradient algorithm. SIAM J. Numer. Anal. **15**(6), 1247–1257 (1978)
6. Perry, J. M.: A class of conjugate gradient algorithms with a two-step variable-metric memory. Discussion Paper 269, Center for Mathematical Studies in Economics and Management Sciences, Northwestern University, Evanston, Illinois (1977)
7. Dai, Y.H., Yuan, Y.X.: A nonlinear conjugate gradient method with a strong global convergence property. SIAM J. Optim. **10**(1), 177–182 (1999)
8. Dai, Y.H., Kou, C.X.: A nonlinear conjugate gradient algorithm with an optimal property and an improved Wolfe line search. SIAM J. Optim. **23**(1), 296–320 (2013)
9. Hager, W.W., Zhang, H.C.: A new conjugate gradient method with guaranteed descent and an efficient line search. SIAM J. Optim. **16**(1), 170–192 (2005)
10. Zhang, L., Zhou, W.J., Li, D.H.: Global convergence of a modified Fletcher-Reeves conjugate gradient method with Armijo-type line search. Numer. Math. **104**, 561–572 (2006)
11. Liu, H.W., Liu, Z.X.: An efficient Barzilai–Borwein conjugate gradient method for unconstrained optimization. J. Optim. Theory Appl. **181**(2), 608–633 (2019)
12. Dai, Y.H., Han, J.Y., Liu, G.H., et al.: Convergence properties of nonlinear conjugate gradient methods. SIAM J. Optim. **10**(2), 345–358 (1999)
13. Dai, Y.H., Liao, L.Z.: New conjugacy conditions and related nonlinear conjugate gradient methods. Appl. Math. Optim. **43**(1), 87–101 (2001)
14. Hager, W.W., Zhang, H.C.: A survey of nonlinear conjugate gradient methods. Pac. J. Optim. **2**(1), 35–58 (2006)
15. Dai, Y.H., Yuan, Y.X.: Nonlinear Conjugate Gradient Methods. Shanghai Scientific and Technical Publishers, Shanghai (2000)
16. Hager, W.W., Zhang, H.C.: The limited memory conjugate gradient method. SIAM J. Optim. **23**(4), 2150–2168 (2013)
17. Schmidt, E.: Über die Auflösung linearer Gleichungen mit Unendlich vielen unbekannten. Rend. Circ. Mat. Palermo. Ser. **1**(25), 53–77 (1908)
18. Liu, D.C., Nocedal, J.: On the limited memory BFGS method for large scale optimization. Math. Program. **45**(1–3), 503–528 (1989)
19. Nocedal, J., Wright, S.J.: Numerical Optimization. Springer, New York (1999)
20. Hager, W.W., Zhang, H.C.: Algorithm 851:conjugate gradient CG_DESCENT, a conjugate gradient method with guaranteed descent. ACM Trans. Math. Softw. **32**(1), 113–137 (2006)
21. Biglari, F., Hassan, M.A., Leong, W.J.: New quasi-Newton methods via higher order tensor models. J. Comput. Appl. Math. **235**(8), 2412–2422 (2011)

22. Wei, Z.X., Li, G.Y., Qi, L.Q.: New quasi-Newton methods for unconstrained optimization problems. Appl. Math. Comput. **175**(2), 1156–1188 (2006)
23. Li, D.H., Fukushima, M.: On the global convergence of BFGS method for nonconvex unconstrained optimization problems. SIAM J. Optim. **11**(4), 1054–1064 (2001)
24. Yuan, Y.X.: A modified BFGS algorithm for unconstrained optimization. IMA J. Numer. Anal. **11**(3), 325–332 (1991)
25. Dai, Y.H., Yuan, J.Y., Yuan, Y.X.: Modified two-point stepsize gradient methods for unconstrained optimization problems. Comput. Optim. Appl. **22**(1), 103–109 (2002)
26. Liu, Z.X., Liu, H.W.: An efficient gradient method with approximate optimal stepsize for large-scale unconstrained optimization. Numer. Algorithms **78**(1), 21–39 (2018)
27. Liu, Z.X., Liu, H.W.: An efficient gradient method with approximately optimal stepsize based on tensor model for unconstrained optimization. J. Optim. Theory Appl. **181**(2), 608–633 (2019)
28. Tarzanagh, D.A., Reza Peyghami, M.: A new regularized limited memory BFGS-type method based on modified secant conditions for unconstrained optimization problems. J. Glob. Optim. **63**, 709–728 (2015)
29. Barzilai, J., Borwein, J.M.: Two-point step size gradient methods. IMA J. Numer. Anal. **8**(1), 141–148 (1988)
30. Andrei, N.: Open problems in nonlinear conjugate gradient algorithms for unconstrained optimization. Bull. Malays. Math. Sci. Soc. **34**(2), 319–330 (2011)
31. Gould, N.I.M., Orban, D., Toint, P.L.: CUTEr and SifDec: a constrained and unconstrained testing environment, revisited. ACM Trans. Math. Softw. **29**(4), 373–394 (2003)
32. Dolan, E.D., More, J.J.: Benchmarking optimization software with performance profiles. Math. Program. **91**(2), 201–213 (2002)

Affiliations

Zexian Liu[1,2] · Hongwei Liu[1] · Yu-Hong Dai[2]

Zexian Liu
liuzx@lsec.cc.ac.cn

Hongwei Liu
hwliu@mail.xidian.edu.cn

[1] School of Mathematics and Statistics, Xidian University, Xi'an 710126, China

[2] LSEC, ICMSEC, Academy of Mathematics and Systems Science, Chinese Academy of Sciences, Beijing 100190, China