A Nonlinear Conjugate Gradient Algorithm with An Optimal Property and An Improved Wolfe Line Search*

Yu-Hong Dai and Cai-Xia Kou

State Key Laboratory of Scientific and Engineering Computing,
Institute of Computational Mathematics and Scientific/Engineering Computing,
AMSS, Chinese Academy of Sciences, Beijing 100190, CHINA
Email addresses: {dyh, koucx}@lsec.cc.ac.cn

Abstract

In this paper, we seek the conjugate gradient direction closest to the direction of the scaled memoryless BFGS method and propose a family of conjugate gradient methods for unconstrained optimization. An improved Wolfe line search is also proposed, which can avoid a numerical drawback of the Wolfe line search and guarantee the global convergence of the conjugate gradient method under mild conditions. To accelerate the algorithm, we introduce adaptive restarts along negative gradients based on how the function is close to some quadratic function during some previous iterations. Numerical experiments with the CUTEr collection show that the proposed algorithm is promising.

Key words: conjugate gradient method, memoryless BFGS method, unconstrained optimization, global convergence, Wolfe line search.

1 Introduction

Consider the unconstrained optimization problem

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

where f is smooth and its gradient g is available. More exactly, we assume that f satisfies

Assumption 1.1. *(i) f is bounded below; namely, $f(x) > -\infty$ for all $x \in \mathcal{R}^n$; (ii) f is differentiable and its gradient g is Lipschitz continuous; namely, there exists a constant $L > 0$ such that*

$$\|\nabla f(x) - \nabla f(y)\| \le L\|x - y\| \quad for \quad any \quad x, y \in \mathcal{R}^n, \tag{1.2}$$

where $\|\cdot\|$ stands for the Euclidean norm.

Conjugate gradient methods are very useful for solving (1.1), especially if its dimension n is large. They are of the form

$$x_{k+1} = x_k + \alpha_k d_k, \tag{1.3}$$

[*] This work was partly supported by the Chinese NSF grants (no. 10831106 and 10971017), the CAS grant (no. kjcx-yw-s7-03) and the China National Funds for Distinguished Young Scientists (no. 11125107).

where the stepsize $\alpha_k > 0$ is obtained by some line search. The next search direction d_{k+1} $(k \geq 1)$ is generated by

$$d_{k+1} = -g_{k+1} + \beta_k\, d_k, \tag{1.4}$$

where $g_{k+1} = \nabla f(x_{k+1})$ and $d_1 = -g_1$. The scalar $\beta_k \in \mathcal{R}$ is so chosen that (1.3)-(1.4) reduces to the linear conjugate gradient method if f is a strictly convex quadratic function and if α_k is the exact one-dimensional minimizer. For general nonlinear functions, different choices of β_k lead to different conjugate gradient methods. Well-known formulae for β_k are called the Fletcher-Reeves (FR), Hestenes-Stiefel (HS), Polak-Ribière-Polyak (PRP) and Dai-Yuan (DY) formulae (see [14]; [20]; [29], [30] and [8], respectively), and are given by

$$\begin{aligned} \beta_k^{FR} &= \frac{\|g_{k+1}\|^2}{\|g_k\|^2}, & \beta_k^{HS} &= \frac{g_{k+1}^T y_k}{d_k^T y_k}, \\ \beta_k^{PRP} &= \frac{g_{k+1}^T y_k}{\|g_k\|^2}, & \beta_k^{DY} &= \frac{\|g_{k+1}\|^2}{d_k^T y_k}, \end{aligned}$$

where $y_k = g_{k+1} - g_k$.

Recent efforts have been made to relate the nonlinear conjugate gradient method to modified conjugacy conditions. Specifically, Dai and Liao [7] considered the following conjugacy condition

$$d_{k+1}^T y_k = -t\, g_{k+1}^T s_k, \tag{1.5}$$

where $s_k = \alpha_k d_k = x_{k+1} - x_k$ and t is some parameter, and derived a new formula for β_k

$$\beta_k^{DL}(t) = \frac{g_{k+1}^T y_k}{d_k^T y_k} - t\, \frac{g_{k+1}^T s_k}{d_k^T y_k}. \tag{1.6}$$

If f is a convex quadratic function and $g_{k+1}^T s_k \neq 0$, it was shown in [7] that a one-dimensional minimizer along the corresponding direction of (1.6) with a small $t > 0$ will lead to a bigger descent than that brought with $t = 0$. Due to the existence of the parameter t, it would be more suitable to call the methods (1.3), (1.4) and (1.6) by Dai-Liao family of conjugate gradient methods (one can see [4] and the references therein for more families of conjugate gradient methods). Further, Dai and Liao [7] considered the following truncated form of (1.6),

$$\beta_k^{DL+}(t) = \max\left\{\frac{g_{k+1}^T y_k}{d_k^T y_k},\, 0\right\} - t\, \frac{g_{k+1}^T s_k}{d_k^T y_k}. \tag{1.7}$$

Despite of possible negative values of β_k^{DL+}, we still use the sign + to symbolize *truncation* in order to remember the truncation introduced by Powell [33] and analyzed by Gilbert and Nocedal [15] for the PRP method (they considered $\beta_k^{PRP+} = \max\{\beta_k^{PRP},\, 0\}$).

Hager and Zhang [17] paid attention to the self-scaling memoryless BFGS method by Perry [28] and Shanno [34] and proposed the formula

$$\beta_k^N = \frac{g_{k+1}^T y_k}{d_k^T y_k} - 2\frac{\|y_k\|^2}{d_k^T y_k}\frac{g_{k+1}^T d_k}{d_k^T y_k}, \tag{1.8}$$

which can be regarded as (1.6) with $t = \frac{2\|y_k\|^2}{s_k^T y_k}$. Interestingly enough, they were able to establish for their method the sufficient descent condition

$$-g_k^T d_k \geq \frac{7}{8}\|g_k\|^2, \quad \forall\, k \geq 1, \tag{1.9}$$

provided that $d_k^T y_k \neq 0$. To establish global convergence for general nonlinear functions, they considered the following truncated form

$$\bar{\beta}_k^N = \max\left\{\beta_k^N, \frac{-1}{\|d_k\| \min\{\eta, \|g_k\|\}}\right\}, \tag{1.10}$$

where $\eta > 0$ is a constant. A fortran code, called cg_descent, was also built based on the formula (1.10) and the so-called approximate Wolfe line search (see [18]). In a survey paper [19], the authors introduced a parameter $\theta_k \geq \frac{1}{4}$ in (1.8), yielding

$$\beta_k^{HZ}(\theta_k) = \frac{g_{k+1}^T y_k}{d_k^T y_k} - \theta_k \frac{\|y_k\|^2}{d_k^T y_k} \frac{g_{k+1}^T d_k}{d_k^T y_k} \tag{1.11}$$

(see the relation (7.1) in [19]). Due to the existence of the parameter θ_k, it would be more convenient to call the methods (1.3), (1.4) and (1.11) by Hager-Zhang family of conjugate gradient methods. It is obvious that β_k^N is corresponding to $\beta_k^{HZ}(\theta_k)$ with $\theta_k \equiv 2$. In this paper, we will show that the scheme in the Hager-Zhang family corresponding to $\theta_k \equiv 1$ performs better than both the $\theta_k \equiv 2$ scheme and all the other schemes that were tested. We will provide both numerical and theoretical justification for the better performance associated with the choice $\theta_k \equiv 1$.

More recent advances can be found in Yabe and Takano [39] and Li *et al.* [21], who studied conjugate gradient methods based on two variants of the conjugacy condition (1.5); namely, by replacing y_k with more efficient vectors. Some generalizations of the Hager-Zhang family (1.11) were provided by Yu *et al.* [40, 41]. The works by Cheng and Liu [3] and Zhang *et al.* [42] investigated new conjugate gradient methods that can ensure the sufficient descent property, namely, $-g_k^T d_k \geq c\|g_k\|^2$ for some constant $c > 0$ and all $k \geq 1$. More recent reviews on nonlinear conjugate gradient methods can be found in Dai [5] and Hager and Zhang [19].

One main contribution of this paper is to seek the conjugate gradient direction that is closest to the direction of the scaled memoryless BFGS method, providing the following family of conjugate gradient methods for unconstrained optimization

$$\beta_k(\tau_k) = \frac{g_{k+1}^T y_k}{d_k^T y_k} - \left(\tau_k + \frac{\|y_k\|^2}{s_k^T y_k} - \frac{s_k^T y_k}{\|s_k\|^2}\right) \frac{g_{k+1}^T s_k}{d_k^T y_k}, \tag{1.12}$$

where τ_k is a parameter corresponding to the scaling parameter in the scaled memoryless BFGS method. Among many others, four different choices of τ_k are analyzed and tested with the following truncation,

$$\beta_k^+(\tau_k) = \max\left\{\beta_k(\tau_k), \eta \frac{g_{k+1}^T d_k}{\|d_k\|^2}\right\}, \tag{1.13}$$

where $\eta \in [0, 1)$ is some parameter. We found that the most efficient choice is corresponding to

$$\tau_k = \frac{s_k^T y_k}{\|s_k\|^2}, \tag{1.14}$$

which was dated back to Oren and Luenberger [25, 26]. Surprisingly enough, in this case, substituting (1.14) into (1.12) gives the following very simple formula,

$$\beta_k = \frac{g_{k+1}^T y_k}{d_k^T y_k} - \frac{\|y_k\|^2}{s_k^T y_k} \frac{g_{k+1}^T s_k}{d_k^T y_k}, \tag{1.15}$$

which is corresponding to the Dai-Liao family of methods (1.6) with $t = \frac{\|y_k\|^2}{s_k^T y_k}$. It is also a special member of the Hager-Zhang family of methods (1.11) with $\theta_k \equiv 1$. More efficient choices of τ_k in (1.12) still remains under investigation.

The rest of this paper is organized as follows. In the next section, we will seek the conjugate gradient direction that is closest to the direction of the scaled memoryless BFGS method and propose a family of conjugate gradient methods for unconstrained optimization. In Section 3, we discuss how to choose the stepsize α_k in (1.3). This is also an important issue in nonlinear conjugate gradient methods. Specifically, we will provide a new strategy for the choice of the initial stepsize (see Algorithm 3.1) and develop an improved Wolfe line search (see (3.6) and (3.4), or Algorithm 3.2). In Section 4, we will present our conjugate gradient algorithm, Algorithm 4.1, which is combined with dynamic restarts. Meanwhile, global convergence results of the algorithm with or without restarts are established under the improved Wolfe line search. In Section 5, we compare the Dolan-Moré [11] performance profile of the new algorithm with cg_descent by Hager and Zhang [18] and test the efficiency of the new restart technique using the unconstrained optimization problems from the CUTEr collection. Conclusions and discussions are made in the last section.

2 A New Family of Conjugate Gradient Methods

The aim of this section is to derive a new family of conjugate gradient methods from the self-scaling memoryless BFGS method by Perry [28] and Shanno [34], which defines the search direction by

$$d_{k+1} = -H_{k+1}\, g_{k+1}, \tag{2.1}$$

where

$$H_{k+1} = \frac{1}{\tau_k}\left(I - \frac{s_k y_k^T + y_k s_k^T}{s_k^T y_k}\right) + \left(1 + \frac{1}{\tau_k}\frac{\|y_k\|^2}{s_k^T y_k}\right)\frac{s_k s_k^T}{s_k^T y_k}, \tag{2.2}$$

where τ_k is a scaling parameter. The approximation matrix H_{k+1} can be regarded to obtain from a scaled identity matrix $\frac{1}{\tau_k} I$ by the BFGS updating formula. Substituting (2.2) into (2.1) leads to the search direction with a multiplier difference

$$d_{k+1}^{PS} = -g_{k+1} + \left[\frac{g_{k+1}^T y_k}{s_k^T y_k} - \left(\tau_k + \frac{\|y_k\|^2}{s_k^T y_k}\right)\frac{g_{k+1}^T s_k}{s_k^T y_k}\right] s_k + \frac{g_{k+1}^T s_k}{s_k^T y_k}\, y_k. \tag{2.3}$$

Noting that $s_k = \alpha_k\, d_k$, the simple deletion of the last term in (2.3) leads to the conjugate gradient method

$$d_{k+1}^{D} = -g_{k+1} + \beta_k^D(\tau_k)\, d_k, \tag{2.4}$$

where

$$\beta_k^D(\tau_k) = \frac{g_{k+1}^T y_k}{d_k^T y_k} - \left(\tau_k + \frac{\|y_k\|^2}{s_k^T y_k}\right)\frac{g_{k+1}^T s_k}{d_k^T y_k}. \tag{2.5}$$

Particularly, if τ_k is chosen to be the value suggested by Oren and Spedicato [26],

$$\tau_k^H = \frac{\|y_k\|^2}{s_k^T y_k}, \tag{2.6}$$

the formula (2.5) reduces to (1.8), which is provided by Hager and Zhang [17].

We are interested in more efficient conjugate gradient variants arising from (2.3) based on the following two observations. Firstly, there are more efficient ways to choose the scaling parameter τ_k. Oren and Luenberger [25, 26] proposed the scaling parameter $\frac{s_k^T y_k}{s_k^T B_k s_k}$ with $B_k = H_k^{-1}$ for the BFGS quasi-Newton method (see the relation (9) in [25]). If H_k is the identity matrix, this choice reduces to

$$\tau_k^B = \frac{s_k^T y_k}{\|s_k\|^2}. \tag{2.7}$$

Al-Baali [1] suggested to choose

$$\bar{\tau}_k^H = \min\left\{1, \frac{\|y_k\|^2}{s_k^T y_k}\right\} \quad \text{and} \quad \bar{\tau}_k^B = \min\left\{1, \frac{s_k^T y_k}{\|s_k\|^2}\right\} \tag{2.8}$$

(see the relations (30) and (31) in [1]). For more choices on scalar τ_k, we refer readers to [1, 25, 26, 27] and the references therein.

Secondly, there is a more reasonable way to deal with the last term in (2.3) instead of simple deletion. Specifically, denoting the one-dimensional manifold

$$\mathcal{S}_{k+1} = \{-g_{k+1} + \beta\, d_k : \beta \in \mathcal{R}\}, \tag{2.9}$$

we can choose the vector in $\mathcal{S}_{k+1}$ closest to d_{k+1}^{PS} in (2.3) as the next search direction; namely,

$$d_{k+1}^P = \arg\min \left\{\|d - d_{k+1}^{PS}\|_2 : d \in \mathcal{S}_{k+1}\right\}. \tag{2.10}$$

Noting that the value $\zeta = \frac{d_k^T y_k}{\|d_k\|^2}$ minimizes $\|y_k - \zeta\, d_k\|$ for $\zeta \in \mathcal{R}$, we can deduce that the search direction in (2.10) is equivalent to

$$d_{k+1}^P = -g_{k+1} + \beta_k(\tau_k)\, d_k, \tag{2.11}$$

where

$$\beta_k(\tau_k) = \frac{g_{k+1}^T y_k}{d_k^T y_k} - \left(\tau_k + \frac{\|y_k\|^2}{s_k^T y_k} - \frac{s_k^T y_k}{\|s_k\|^2}\right) \frac{g_{k+1}^T s_k}{d_k^T y_k}. \tag{2.12}$$

If $d_k^T g_{k+1} = 0$, the second term in (2.12) is missing and reduces to the HS or PRP formula. Therefore we have obtained a family of conjugate gradient methods (1.3), (2.11) and (2.12), where the parameter τ_k is corresponding to the scaling parameter in the self-scaling memoryless BFGS method.

It is interesting to note that the formula (2.12) is corresponding to (1.6) if we adaptively choose t to be

$$t_k = \tau_k + \frac{\|y_k\|^2}{s_k^T y_k} - \frac{s_k^T y_k}{\|s_k\|^2}. \tag{2.13}$$

To establish a basic property for the family of conjugate gradient methods (1.3), (2.11) and (2.12), we define

$$p_k = \frac{\|d_k\|^2 \|y_k\|^2}{(d_k^T y_k)^2} \quad \text{and} \quad \gamma_k = \tau_k \frac{\|s_k\|^2}{s_k^T y_k}. \tag{2.14}$$

Lemma 2.1. *For the family of conjugate gradient methods* (1.3), (2.11) *and* (2.12), *if* $d_k^T y_k \neq 0$, *we always have that*

$$-d_{k+1}^T g_{k+1} \geq \min\left(\gamma_k, \frac{3}{4}\right) \|g_{k+1}\|^2. \tag{2.15}$$

Proof. Noting that $s_k = \alpha_k\, d_k$, we can write the search direction d_{k+1} in the form

$$d_{k+1}^P = -H_{k+1}^P g_{k+1}, \tag{2.16}$$

where

$$H_{k+1}^P = I - \frac{d_k z_k^T}{d_k^T y_k}, \quad z_k = y_k - p_k\, s_k. \tag{2.17}$$

To proceed our analysis, we symmetrize H_{k+1}^P and define

$$\overline{H}_{k+1}^P = \frac{1}{2}\left[H_{k+1}^P + \left(H_{k+1}^P\right)^T\right] = I - \frac{d_k z_k^T + z_k d_k^T}{2\, d_k^T y_k}. \tag{2.18}$$

For any vectors $u, v \in \mathcal{R}^n$, notice that

$$\left(uv^T + vu^T\right)\left(u \pm \frac{\|u\|}{\|v\|}v\right) = \left(u^T v \pm \|u\|\,\|v\|\right)\left(u \pm \frac{\|u\|}{\|v\|}v\right). \tag{2.19}$$

By this, it is not difficult to see that the minimal eigenvalue of $\overline{H}_{k+1}^P$ is

$$\lambda_{\min} = \min\left\{1,\, 1 - \frac{1}{2}\left(\frac{d_k^T z_k}{d_k^T y_k} + \frac{\|d_k\|\,\|z_k\|}{|d_k^T y_k|}\right)\right\}. \tag{2.20}$$

With the definitions of p_k and γ_k, we can rewrite (2.20) as

$$\lambda_{\min} = \min\left\{1,\, \frac{1}{2}\left(p_k + \gamma_k - \sqrt{p_k^2 + (2\gamma_k - 3)\,p_k + \left(\gamma_k^2 - 4\gamma_k + 3\right)}\right)\right\}. \tag{2.21}$$

Now we consider the second term in the braces of (2.21). It is easy to verify that if $\gamma_k \le \frac{3}{4}$, it is monotonically increasing for $p_k \in [1, +\infty)$ and hence reaches its minimum, that is γ_k, at $p_k = 1$; if $\gamma_k > \frac{3}{4}$, it is monotonically decreasing for $p_k \in [1, +\infty)$ and hence is always greater than its limit $\frac{3}{4}$ as p_k tends to $+\infty$. Thus we always have

$$-g_{k+1}^T d_{k+1} = g_{k+1}^T \overline{H}_{k+1}^P g_{k+1} \ge \lambda_{\min}\,\|g_{k+1}\|^2 \ge \min\left(\gamma_k,\, \frac{3}{4}\right)\,\|g_{k+1}\|^2, \tag{2.22}$$

which completes our proof. □

Lemma 2.2. *Assume that f satisfies Assumption 1.1. Consider the family of conjugate gradient methods* (1.3), (2.11) *and* (2.12). *If τ_k is chosen to be any of τ_k^B, τ_k^H, $\bar{\tau}_k^B$ and $\bar{\tau}_k^H$ and if $d_k^T y_k \ne 0$, we have that*

$$-g_{k+1}^T d_{k+1} \ge c\,\|g_{k+1}\|^2 \quad \textit{for some positive constant } c > 0. \tag{2.23}$$

Proof. (i) If $\tau_k = \tau_k^B$, we have by (2.14) that $\gamma_k = 1$, which with Lemma 2.1 implies the truth of (2.23) with $c = \frac{3}{4}$; (ii) If $\tau_k = \tau_k^H$, then $\gamma_k = p_k$. By (2.21) and the fact that $p_k \ge 1$, we see that

$$\lambda_{\min} = \min\left\{1,\, p_k - \sqrt{p_k^2 - \frac{7}{4}p_k + \frac{3}{4}}\right\} > \min\left\{1,\, p_k - \sqrt{p_k^2 - \frac{7}{4}p_k + \frac{49}{64}}\right\} = \frac{7}{8}. \tag{2.24}$$

So (2.23) holds with $c = \frac{7}{8}$; (iii) By the Lipschitz condition (1.2) and the definitions of y_k and s_k, we have that

$$\|y_k\| \leq L\,\|s_k\|. \tag{2.25}$$

If $\frac{s_k^T y_k}{\|s_k\|^2} < 1$, we have $\tau_k = \frac{s_k^T y_k}{\|s_k\|^2}$ and hence $\gamma_k = 1$. Otherwise, if $\frac{s_k^T y_k}{\|s_k\|^2} \geq 1$, we must have from this, the Cauchy-Schwartz inequality and (2.25) that

$$\gamma_k = \frac{\|s_k\|^2}{s_k^T y_k} \geq \frac{\|s_k\|}{\|y_k\|} \geq \frac{1}{L}. \tag{2.26}$$

Consequently, we always have $\gamma_k \geq \min(1, \frac{1}{L})$. By Lemma 2.1, (2.23) holds with $c = \min(\frac{3}{4}, \frac{1}{L})$; (iv) If $\frac{\|y_k\|^2}{s_k^T y_k} < 1$, we have that $\gamma_k = p_k \geq 1$. Otherwise, if $\frac{\|y_k\|^2}{s_k^T y_k} \geq 1$, we know that $s_k^T y_k > 0$ and the inequality (2.26) is still valid. Thus we also have $\gamma_k \geq \min(1, \frac{1}{L})$. By Lemma 2.1, (2.23) holds with $c = \min(\frac{3}{4}, \frac{1}{L})$. □

To generalize Lemma 2.2, we consider the convex combination of τ_k^H and τ_k^B

$$\tau_k = \nu \frac{\|y_k\|^2}{s_k^T y_k} + (1-\nu)\frac{s_k^T y_k}{\|s_k\|^2}, \tag{2.27}$$

where $\nu \in [0, 1]$. This formed interval of τ_k is corresponding to the subclass of the self-scaled variable metric (SSVM) methods with the scaling parameter in $\left[\frac{s_k^T y_k}{y_k^T H_k y_k}, \frac{s_k^T H_k^{-1} s_k}{s_k^T y_k}\right]$ (assuming $s_k^T y_k > 0$). This subclass was proposed in [24] and it forms the basis for the SSVM algorithms in [25], [26] and [27]. Specially, [27] studied on this subclass of SSVM algorithms with some additional optimal property.

Lemma 2.3. *Assume that f satisfies Assumption 1.1. Consider the subfamily of conjugate gradient methods* (1.3), (2.11) *and* (2.12), *where τ_k is of the form (2.27) with $\nu \in [0, 1]$. If $d_k^T y_k \neq 0$, we have that $-g_{k+1}^T d_{k+1} \geq \frac{3}{4}\,\|g_{k+1}\|^2$.*

Proof. Since, by (2.12), $\beta_k(\tau_k)$ is linear with τ_k, it is easy see that d_{k+1} and hence $-g_{k+1}^T d_{k+1}$ is also linear with τ_k. By items (i) and (ii) of the proof to Lemma 2.2, we know that (2.23) holds with $c = \frac{3}{4}$ for both τ_k^H and τ_k^B. Thus the statement is true for their convex combination. □

Powell [32] constructed a counter-example showing that the PRP method with exact line search may not converge for general nonlinear functions. Since for any τ_k, $\beta_k(\tau_k) = \beta_k^{PRP}$ if $g_{k+1}^T d_k = 0$, Powell's example can also be used to show the method (1.3) and (2.11) with $\beta_k(\tau_k)$ given by (2.12) need not converge for general functions. Therefore similarly to Gilbert and Nocedal [15], who proved the global convergence of the PRP method for general functions by restricting $\beta_k \geq 0$, we replace (2.12) by

$$\beta_k^+(\tau_k) = \max\left\{\beta_k(\tau_k),\, \eta\frac{g_{k+1}^T d_k}{\|d_k\|^2}\right\}, \tag{2.28}$$

where $\eta \in [0, 1)$ is some parameter and its suggested value is 0.5 in our practical computations. This way of truncation comes from the observation that, while projecting the d_{k+1}^{PS} in (2.3) into the one-dimensional manifold (2.9), its last term provides the contribution $\frac{g_{k+1}^T d_k}{\|d_k\|^2}\, d_k$. Further, the following downhill direction

$$-g_{k+1} + \eta\, \frac{g_{k+1}^T d_k}{\|d_k\|^2}\, d_k \tag{2.29}$$

seems to be a better restart direction than $-g_{k+1}$ since it includes some curvature information achieved along the previous search direction.

Lemma 2.4. *Assume that f satisfies Assumption 1.1. Consider the family of conjugate gradient methods* (1.3), (2.11) *and* (2.12), *where $\beta_k(\tau_k)$ is replaced with the $\beta_k^+(\tau_k)$ in* (2.28) *and τ_k is chosen to be any of τ_k^B, τ_k^H, $\bar{\tau}_k^B$ and $\bar{\tau}_k^H$. If $d_k^T y_k \neq 0$, we have that*

$$-g_{k+1}^T d_{k+1} \geq \bar{c}\, \|g_{k+1}\|^2 \quad \textit{for some positive constant } \bar{c} > 0. \tag{2.30}$$

Proof. By Lemma 2.2, we only need to consider the case that

$$\beta_k^+(\tau_k) = \eta \frac{g_{k+1}^T d_k}{\|d_k\|^2} \quad \text{with} \quad 0 \leq \eta < 1.$$

In this case, it is obvious that

$$-d_{k+1}^T g_{k+1} = \|g_{k+1}\|^2 - \eta \frac{(g_{k+1}^T d_k)^2}{\|d_k\|^2} \geq (1-\eta)\|g_{k+1}\|^2.$$

This, with Lemma 2.2, indicates that (2.30) holds with $\bar{c} = \min\left(c, (1-\eta)\right)$. □

Remark 1. In the above, we have proposed a family of conjugate gradient methods (1.3), (2.11) and (2.12). Its proposition is natural; namely, by projecting the self-scaling memoryless BFGS direction by Perry [28] and Shanno [34] into the one-dimensional manifold (2.9). Four choices for the parameter τ_k are presented and analyzed. The numerical experiments in Section 5 will suggest that the τ_k^B in (2.7) is the most efficient one. If we substitute this special choice into (2.12) and (2.28), we can obtain a relatively simple formula for β_k and its truncation form. They are

$$\beta_k = \frac{g_{k+1}^T y_k}{d_k^T y_k} - \frac{\|y_k\|^2}{s_k^T y_k} \frac{g_{k+1}^T s_k}{d_k^T y_k} \tag{2.31}$$

and

$$\beta_k^+ = \max\left\{ \frac{g_{k+1}^T y_k}{d_k^T y_k} - \frac{\|y_k\|^2}{d_k^T y_k} \frac{g_{k+1}^T d_k}{d_k^T y_k},\ \eta \frac{g_{k+1}^T d_k}{\|d_k\|^2} \right\}, \tag{2.32}$$

where $\eta \in [0,\, 1)$. We see that the formula (2.31) is corresponding to the Dai-Liao family of methods (1.6) with $t = \frac{\|y_k\|^2}{s_k^T y_k}$. (2.31) also differs (1.8) only with a constant coefficient in the second term and corresponds with the Hager-Zhang family of methods (1.11) with $\theta_k \equiv 1$.

Remark 2. The interval of τ_k formed in (2.27) with $\nu \in [0,\, 1]$ gives a subfamily of conjugate gradient methods with

$$\frac{g_{k+1}^T y_k}{d_k^T y_k} - \theta_k \frac{\|y_k\|^2}{s_k^T y_k} \frac{g_{k+1}^T s_k}{d_k^T y_k}, \tag{2.33}$$

where

$$\theta_k \in \left[1,\, 2 - \frac{(s_k^T y_k)^2}{\|s_k\|^2\, \|y_k\|^2}\right]. \tag{2.34}$$

Surprisingly, this subfamily has (2.31) as its special member but excludes the Hager-Zhang choice (1.8). This, to some extent, explains the efficiency of the formula (2.31) over (1.8) in our numerical experiments.

Remark 3. Powell [32]'s counter-example was extended in [6] to show that, for any small constant $\varepsilon > 0$, the modified PRP method with $\beta_k = \max\{\beta_k^{PRP},\, -\varepsilon\}$ need not converge for general functions. This implies to some extent that the restriction $\beta_k \geq 0$ is essential in ensuring the global convergence of the PRP method. However, Gilbert and Nocedal [15] showed that, for the PRP method using exact line searches, it is possible that $\beta_k^{PRP} < -\beta_k^{FR} < 0$ for a strongly convex function, although it is known that the original PRP method converges globally in this case. There have been several strategies that allow negative values of β_k and guarantee global convergence of the conjugate gradient method for general functions. For example, Dai and Liao [7] considered the truncation in (1.7) and Hager and Zhang suggested the truncation in (1.10). The restriction (2.28) provides another possibility and proves very useful in our convergence analysis and practical calculations.

3 An Improved Wolfe Line Search

As is known, the search direction and the line search are two important factors of a line search algorithm. The purpose of this section is to develop an improved Wolfe line search, which allows a small increase on the objective function value and can avoid a numerical drawback of the Wolfe line search. As shown in the next section, this improved Wolfe line search guarantees the global convergence of the conjugate gradient method. A strategy is also designed for the choice of the initial stepsize.

For convenience, we denote the one-dimensional line search function to be

$$\phi_k(\alpha) = f(x_k + \alpha\, d_k), \quad \alpha \geq 0. \tag{3.1}$$

The choice of the initial stepsize is important for a line search. For Newton-like methods, the choice $\alpha_k^{(0)} = 1$ is essential in giving rapid convergence rate. For conjugate gradient methods, it is important to use the current information about the problem to make an initial guess [23]. There have been quite a few ways to choose the initial stepsize in the conjugate gradient method, for example, see [12, 35, 23, 17]. However, it does not reach a consensus which one is the best. Specifically, Hager and Zhang [17] chose the stepsize as follows:

$$\alpha_k^{(0)} = \begin{cases} \arg\min\ q(\phi_k(0),\, \phi_k'(0),\, \phi_k(\psi_1\, \alpha_{k-1})), & \text{if } \phi(\psi_1 \alpha_{k-1}) \leq \phi(0); \\ \qquad\qquad \psi_2\, \alpha_{k-1}, & \text{otherwise}, \end{cases} \tag{3.2}$$

where ψ_1 and ψ_2 are positive parameters and $q(\phi_k(0)), \phi_k'(0), \phi_k(\psi_1 \alpha_{k-1}))$ denotes the interpolation function by the three values $\phi_k(0)$, $\phi_k'(0)$ and $\phi_k(\psi_1 \alpha_{k-1})$. The suggested values are $\psi_1 = 0.1$ and $\psi_2 = 2$. In the following, we propose another way for the choice of the initial stepsize.

Algorithm 3.1. ***(A strategy for choosing the initial stepsize)***

Step 0. Given positive parameters ϵ_1, ϵ_2 and ψ;

Step 1. Set $\alpha_k^{(0)} := \max\{\psi\, \alpha_{k-1},\ -2|f_k - f_{k-1}|/d_k^T g_k\}$ *and calculate* $\phi_k(\alpha_k^{(0)})$*;*

Step 2. If $\frac{|\phi_k(\alpha_k^{(0)})-\phi_k(0)|}{\epsilon_1+|\phi_k(0)|} \le \epsilon_2$*, set* $\alpha_k^{(0)} :=$ *arg* $\min\ q(\phi_k(0),\ \phi_k'(0),\ \phi_k(\alpha_k^{(0)}))$.

In the above algorithm, the condition $\frac{|\phi_k(\alpha_k^{(0)})-\phi_k(0)|}{\epsilon_1+|\phi_k(0)|} \le \epsilon_2$ is used to guarantee that the points $x_k + \alpha_k^{(0)} d_k$ and x_k are not far away from each other and again, $q(\phi_k(0)), \phi_k'(0), \phi_k(\alpha_k^{(0)}))$ denotes the interpolation function by the three values $\phi_k(0)$, $\phi_k'(0)$ and $\phi_k(\alpha_k^{(0)})$. The basic idea of Algorithm 3.1 is that, if the points $x_k + \alpha_k^{(0)} d_k$ and x_k are close to each other, we would like to do an interpolation and take the minimizer of the interpolation function as a new initial stepsize. This has the cost of an extra function evaluation in computing $\phi_k(\alpha_k^{(0)})$, but it is worthwhile. Otherwise, we choose the initial stepsize simply as the auxiliary stepsize $\max\{\psi\alpha_{k-1},\ -2|f_k - f_{k-1}|/d_k^T g_k\}$. Instead of using a small value $\psi = 0.1$ in version 3.0 of cg_descent, we pick up a large value $\psi = 5$ in our algorithm. Our numerical experiments show that, comparing with (3.2), Algorithm 3.1 is more suitable for Algorithm 4.1 (here we should mention that the comparison was mainly made version 3.0 of cg_descent. At the same time, we noticed that the condition $\phi_k(\psi_1\alpha_k) \le \phi(0)$ has been dropped in a new version (version 5.0) of cg_descent (May 1, 2011)).

The design of Algorithm 3.1 is also such that at least one quadratic interpolation will be made when the iteration is close to the solution point and hence asymptotical n-step superlinear convergence is expected to be established for the whole conjugate gradient algorithm. This is because in this case the difference $\phi_k(\alpha_k^{(0)}) - \phi_k(0)$ will eventually become small and the condition of Step 2 of Algorithm 3.1 will be satisfied. Consequently, a quadratic interpolation will always be made for the initial stepsize asymptotically.

Next, we introduce new line search conditions, which can avoid a numerical drawback of the Wolfe conditions and ensure the global convergence of the conjugate gradient algorithm.

To this aim, recall the Wolfe conditions

$$\begin{aligned} \phi_k(\alpha) &\le \phi_k(0) + \delta\, \alpha\, \phi_k'(0), && (3.3)\\ \phi_k'(\alpha) &\ge \sigma\, \phi_k'(0), && (3.4) \end{aligned}$$

where $0 < \delta < \sigma < 1$. The Wolfe conditions (3.3)-(3.4) can be dated back to [37, 38] and was used to analyze nonlinear conjugate gradient methods in [8, 9]. Theoretically, under Assumption 1.1 on f, if d_k is a descent direction, there must exist some stepsize $\alpha_k > 0$ satisfying (3.3)-(3.4).

In practical computations, however, the first Wolfe condition, (3.3), may never be satisfied due to the existence of the numerical errors. Assume that $\alpha_k^* > 0$ is the exact minimizer of $\phi_k(\alpha)$. If $\alpha_k^* d_k$ is too small, we have that $\phi_k(0) - \phi_k(\alpha^*) = O(\|\alpha_k^* d_k\|^2)$. Consequently, $\phi_k(\alpha^*)$ is about the same as $\phi_k(0)$ provided that $\|\alpha_k^* d_k\|$ is on the order of square root of machine precision. It turns out that in this case, it is possible that $\phi_k(\alpha) \ge \phi_k(0)$ for all $\alpha \ge 0$ in numerical sense and hence (3.3) is never satisfied in practical computations. This numerical drawback of the Wolfe conditions was carefully analyzed in [17] with a one-dimensional quadratic function. Theoretically, this failure of the Wolfe line search can occur in a neighborhood of any strict local minimizer if the required tolerance error is tiny enough. Even when using the stopping condition $\|g_k\| \le 10^{-6}$, we have observed this possibility for Problem JENSMP from CUTEr collection, which was originally introduced in [22] and has the form

$$f(x_1, x_2) = [4 - (e^{x_1} + e^{x_2})]^2 + [6 - (e^{2x_1} + e^{2x_2})]^2. \qquad (3.5)$$

We used the conjugate gradient method (1.3), (2.11) and (2.28) with τ_k replaced by (2.7) and α_k calculated by the Wolfe line search. At the $16th$ iteration, we obtained

$$x_{16} = (2.5782521324\text{e-}01,\ 2.5782521393\text{e-}01)^T,\quad d_{16} = (9.2964892641\text{e-}06,\ -2.5552928578\text{e-}06)^T$$

with values

$$f_{16} = 1.24362182\text{e+}02,\quad g_{16}^T d_{16} = -9.2954234226\text{e-}11,\quad \|g_{16}\| = 1.5815277275\text{e-}05.$$

We found that even with 50 trial stepsizes ranging from 1.7e-26 to 1.2e-5, the algorithm failed to find a stepsize along the search along d_{16} such that the first Wolfe condition is satisfied.

To avoid the above numerical drawback of the Wolfe line search, Hager and Zhang [17] suggested a combination of the original Wolfe conditions and the approximate Wolfe conditions that $\sigma\,\phi_k'(0) \le \phi_k'(\alpha) \le (2\delta-1)\,\phi_k'(0)$. Their line search performed well in their numerical tests, but cannot guarantee the global convergence of the algorithm in theory. Given a constant parameter $\epsilon > 0$, a positive sequence $\{\eta_k\}$ satisfying $\sum_{k\ge1}\eta_k < +\infty$ and again parameters δ and σ satisfying $0 < \delta < \sigma < 1$, we propose the following modified Wolfe condition

$$\phi_k(\alpha) \le \phi_k(0) + \min\left\{\epsilon|\phi_k'(0)|,\ \delta\alpha\phi_k'(0) + \eta_k\right\} \tag{3.6}$$

and call the line search satisfying (3.6) and (3.4) by *improved Wolfe line search.* The idea behind the condition (3.6) is that, it allows the stepsizes satisfying (3.3) and hence is an extension of the original Wolfe line search; meanwhile, if the trial point is close to x_k, in which case

$$\phi_k(\alpha) \le \phi_k(0) + \epsilon\,|\phi_k(0)|, \tag{3.7}$$

we switch to require

$$\phi_k(\alpha) \le \phi_k(0) + \delta\alpha\phi_k'(0) + \eta_k \tag{3.8}$$

rather than (3.3). The extra positive term η_k in (3.8) or (3.6) allows a slight increase in the function value and hence is helpful in avoiding the numerical drawback of the first Wolfe line search condition (3.3). At the same time, the condition that the sequence $\{\eta_k\}$ is summable can guarantee the global convergence of the algorithm similarly to the Wolfe line search. Specifically, we set $\eta_k = \frac{1}{k^2}$ in our numerical experiments.

Under Assumption 1.1 on f, if $g_k^T d_k < 0$, it is obvious that there must exist a suitable stepsize satisfying (3.6) and (3.4) since they are weaker than the Wolfe conditions. In the following, we describe a detailed procedure to implement the improved Wolfe line search, where a point x_k and a descent direction d_k are given.

Algorithm 3.2. ***(An Improved Wolfe Line Search)***

```
Step 0. Determine $\alpha_k^{(0)}$ via Algorithm 3.1.
    Set $a_0 = 0$, $\phi_a = f(x_k)$, $\phi_a' = g_k^T d_k$ and $b_0 = M$, where $M$ is some big number.
    Set $t_1 = 1.0$, $t_2 = 0.1$, $\rho > 1$ and $i = 0$.
Step 1. Evaluate $\phi_k(\alpha_k^{(i)})$ and test condition (3.6).
    If (3.6) is satisfied, goto Step 2.
    Set $b_i = \alpha_k^{(i)}$, $\phi_b = \phi_k(\alpha_k^{(i)})$, $\alpha^* = argmin\ q(\phi_a, \phi_a', \phi_b)$ and $t_1 = 0.1t_1$.
    Choose $\alpha_k^{(i+1)} := \min\{\max[\alpha^*, a_i + t_1(b_i - a_i)], b_i - t_2(b_i - a_i)\}$.
    Set $i = i + 1$ and goto Step 1.
```

```
Step 2. Evaluate $\phi'(\alpha_k^{(i)})$ and test condition (3.4).
    If (3.4) is satisfied, return $\alpha_k = \alpha_k^{(i)}$, stop.
    Set $t_1 = 0.1$, $t_2 = 0.1 t_2$. If $b_i = M$, goto Step 3.
    Set $a_i = \alpha_k^{(i)}$, $\phi_a = \phi(\alpha_k^{(i)})$, $\phi'_a = \phi'(\alpha_k^{(i)})$ and $\alpha^* = argmin\ q(\phi_a, \phi'_a, \phi_b)$.
    Choose $\alpha_k^{(i+1)} := \min\{\max[\alpha^*, a_i + t_1(b_i - a_i)], b_i - t_2(b_i - a_i)\}$.
    Set $i = i + 1$, goto Step 1.
Step 3. Set $a_i = \alpha_k^{(i)}$, $\phi_a = \phi(\alpha_k^{(i)})$, $\phi'_a = \phi'(\alpha_k^{(i)})$ and $\alpha_k^{(i+1)} = \rho \alpha_k^{(i)}$.
    Set $i = i + 1$, goto Step 1.
```

We can see that the above procedure is similar, but not identical, to the classical implementation of the Wolfe line search in Fletcher [13]. At first, we determine $\alpha_k^{(0)}$ via Algorithm 3.1 and initialize the interval $[a_0, b_0]$ as $[0, M]$, where M is some big number. Then in the current bracket $[a_i, b_i]$, we choose a new trial stepsize $\alpha_k^{(i+1)}$ to be the minimizer of the quadratic interpolation function $q(\phi_a, \phi'_a, \phi_b)$, but prevent it from being arbitrarily close to the extremes of the interval using the preset factors t_1 and t_2. If $\alpha_k^{(i+1)}$ satisfies (3.6) and (3.4), the line search is terminated with $\alpha_k = \alpha_k^{(i+1)}$. If $\alpha_k^{(i+1)}$ only satisfies (3.6), we update $[a_i, b_i]$ as $[\alpha_k^{(i+1)}, b_i]$; otherwise, we update $[a_i, b_i]$ as $[a_i, \alpha_k^{(i+1)}]$. We should note that in case of updating $[a_i, M]$, $\alpha_k^{(i+1)}$ is chosen to be a multiple of a_i, namely, $\rho\, a_i$ with $\rho > 1$ since M is preset to a very big number and the interpolation in the interval $[a_i, M]$ is likely not to be reliable. On the whole, the procedure will generate a sequence of intervals $[a_i, b_i]$ with properties $[a_{i+1}, b_{i+1}] \subset [a_i, b_i]$ for all i, $|b_i - a_i| \to 0$ and

$$\phi_k(a_i) \le \phi_k(0) + \min\{\epsilon\, |\phi_k(0)|, \delta\, a_i\, \phi'(0) + \eta_k\} \quad \text{but} \quad \phi'_k(a_i) < \sigma \phi'_k(0), \tag{3.9}$$

$$\phi_k(b_i) > \phi_k(0) + \min\{\epsilon\, |\phi_k(0)|, \delta\, b_i\, \phi'(0) + \eta_k\}, \tag{3.10}$$

until a satisfactory stepsize is successfully found.

For the improved Wolfe line search, we can establish the Zoutendijk condition (3.11) (see [43]) all the same.

Lemma 3.3. *Assume that f satisfies Assumption 1.1. Consider the iterative method of the form (1.3) where the direction d_k satisfies $g_k^T d_k < 0$ and the stepsize α_k satisfies* (3.6) *and* (3.4). *Then we have that*

$$\sum_{k \ge 1} \frac{(g_k^T d_k)^2}{\|d_k\|^2} < \infty. \tag{3.11}$$

Proof. It follows from the Lipschitz condition (1.2) and the line search condition (3.4) that

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

Thus we have

$$\alpha_k \ge \frac{\sigma - 1}{L} \frac{g_k^T d_k}{\|d_k\|^2}. \tag{3.12}$$

It follows from (3.6) that

$$f_{k+1} \le f_k + \min\{\epsilon |f_k|, \delta \alpha_k g_k^T d_k + \eta_k\} \le f_k + \delta \alpha_k g_k^T d_k + \eta_k, \tag{3.13}$$

which with (3.12) implies that

$$f_k - f_{k+1} + \eta_k \geq c \frac{(g_k^T d_k)^2}{\|d_k\|^2}, \tag{3.14}$$

where $c = \delta(1-\sigma)/L$. Summing (3.14) over k and noting that $\sum_{k\geq 1} \eta_k < +\infty$ and that f is bounded below, we see that (3.11) holds. □

4 Algorithm and Convergence Analysis

In this section, we present and analyze the whole scheme of our new conjugate gradient algorithms with the improved Wolfe line search. An adaptive restart technique will also be incorporated to accelerate the algorithm.

When the algorithm goes on the k-th iteration, we look back at the line search function ϕ_{k-1} at the $(k-1)$-th iteration. We have at least four function or derivative values of ϕ_{k-1}, which are $\phi_{k-1}(0) = f_{k-1}$, $\phi'_{k-1}(0) = g_{k-1}^T d_{k-1}$, $\phi_{k-1}(\alpha_{k-1}) = f_k$ and $\phi'_{k-1}(\alpha_{k-1}) = g_k^T d_{k-1}$, no matter how the stepsize α_{k-1} is found. By the four values, we can define a quantity indicating how ϕ_{k-1} is close to a quadratic function. The basic idea is to do a quadratic interpolation to get q_{k-1} by imposing the three conditions

$$q_{k-1}(0) = \phi_{k-1}(0), \quad q'_{k-1}(0) = \phi'_{k-1}(0), \quad q'_{k-1}(\alpha_{k-1}) = \phi'_{k-1}(\alpha_{k-1}) \tag{4.1}$$

(later, we will use another denotation to express the interpolation function, for example, q_{k-1} is also denoted by $q(\phi_{k-1}(0), \phi'_{k-1}(0), \phi'_{k-1}(\alpha_{k-1}))$). If the value of this interpolation function at α_{k-1}, namely, $q_{k-1}(\alpha_{k-1})$, is close to the real function value $\phi_{k-1}(\alpha_{k-1})$, we think that ϕ_{k-1} tends to be some quadratic function. More exactly, similarly to the ratio used for adjusting the radius in trust region methods, we define the quantity

$$r_{k-1} = \frac{\phi_{k-1}(0) - \phi_{k-1}(\alpha_{k-1})}{q_{k-1}(0) - q_{k-1}(\alpha_{k-1})}. \tag{4.2}$$

Further, noticing that $\phi_{k-1}(0) = q_{k-1}(0) = f_{k-1}$, $\phi_{k-1}(\alpha_{k-1}) = f_k$ and by direct calculations, $q_{k-1}(\alpha_{k-1}) = f_{k-1} + \frac{1}{2}\, \alpha_{k-1}\, (g_{k-1}^T d_{k-1} + g_k^T d_{k-1})$, the quantity r_{k-1} can be simplified as

$$r_{k-1} = \frac{2\,(f_k - f_{k-1})}{\alpha_{k-1}\,(g_{k-1}^T d_{k-1} + g_k^T d_{k-1})}. \tag{4.3}$$

If r_{k-1} is close to 1, we think that ϕ_{k-1} is close to some quadratic function and otherwise, not. A successful use of this quantity r_{k-1} in designing gradient descent algorithms can be found in [10].

In the nonlinear conjugate gradient field, since it is general that the function is nonlinear at the initial stage and tends to be quadratic when the iterate is close to some solution point, we believe that the quantity r_{k-1} must be very useful in designing nonlinear conjugate gradient algorithms. More exactly, if there are continuously many iterations such that r_k is close 1, we restart the algorithm with the steepest descent direction. In this case, we think that the algorithm is very likely to enter some region where the objective function is close to some quadratic function and hence a restart along $-g_k$ is worthwhile provided that not all values of r_k are around one since the last restart.

In addition, if the number of the total iterations since the last restart reaches some threshold, $MaxRestart$, we also restart the algorithm. In our experiment, we choose this threshold to be $6n$ to avoid frequent restarts for relatively small nonlinear functions. For large-scale problems, this restarting criterion is generally not active.

The following is a detailed description of our algorithm.

Algorithm 4.1. ***(A Nonlinear Conjugate Gradient Algorithm)***

```
Step 0. Given $x_1 \in \mathcal{R}^n$, $\varepsilon > 0$, $\epsilon_4 > 0$ and positive integers $MaxRestart$, $MinQuad$.

Step 1. Set $k := 1$. If $\|g_1\| \le \varepsilon$, stop.
        Let $d_1 = -g_1$ and set $IterRestart := 0$ and $IterQuad := 0$.

Step 2. Compute a stepsize $\alpha_k > 0$ via Algorithm 3.2.

Step 3. Let $x_{k+1} = x_k + \alpha_k d_k$. If $\|g_{k+1}\| \le \varepsilon$, stop.
        $IterRestart := IterRestart + 1$.
        Compute $r_k$ by (4.3). If $|r_k - 1| \le \epsilon_4$, $IterQuad := IterQuad + 1$; else, $IterQuad := 0$.

Step 4. If $IterRestart = MaxRestart$ or ($IterQuad = MinQuad$ and $IterQuad \ne IterRestart$),
        let $d_{k+1} = -g_{k+1}$ and set $IterRestart := 0$, $IterQuad := 0$, $k := k + 1$, goto Step 2.

Step 5. Compute $\beta_k$ by (2.28) and $d_{k+1}$ by (1.4). $k := k + 1$, goto Step 2.
```

Particularly, if the parameter τ_k in (2.28) is chosen to be τ_k^H in (2.6), τ_k^B in (2.7), $\bar{\tau}_k^H$ and $\bar{\tau}_k^B$ in (2.8), the above algorithm is called as Algorithms 4.1(a), 4.1(b), 4.1(c) and 4.1(d), respectively.

Now we analyze the global convergence properties of the above conjugate gradient algorithm. For convenience, assume that

$$g_k \ne 0, \quad \forall\, k \ge 1$$

throughout this section, for otherwise a stationary point has been found. Since a restart along the negative gradient is done in at least $MaxRestart$ iterations, there must be global convergence of Algorithm 4.1 for general functions. Actually, assuming that $d_{k_i} = -g_{k_i}$ for some infinite subsequence $\{k_i\}$, we have from Lemma 3.3 that $\lim\limits_{i\to\infty} \|g_{k_i}\| = 0$. In the following, we consider the global convergence properties of Algorithm 4.1 without any restarts.

For uniformly convex functions, we have the following convergence result.

Theorem 4.2. *Assume that f satisfies Assumption 1.1. Consider the search direction defined by* (1.3), (2.11), (2.12)*, where τ_k is chosen to be any of τ_k^H, τ_k^B, $\bar{\tau}_k^H$ and $\bar{\tau}_k^B$, and where stepsize α_k is calculated by the line search satisfying* (3.6) *and* (3.4)*. If, further, f is uniformly convex, namely, there exists a constant $\mu > 0$ such that*

$$(\nabla f(x) - \nabla f(y))^T (x - y) \ge \mu \|x - y\|^2, \quad \forall\, x, y \in \mathcal{R}^n, \tag{4.4}$$

we have that

$$\lim_{k\to\infty} g_k = 0. \tag{4.5}$$

Proof. It follows from (1.2) and (4.4) that

$$\|y_k\| \le L\, \|s_k\|, \tag{4.6}$$

$$d_k^T y_k \ge \mu\, \|d_k\|\, \|s_k\|. \tag{4.7}$$

By (4.4) and (4.6), it is easy to see that for any τ_k of τ_k^H, τ_k^B, $\bar{\tau}_k^H$ and $\bar{\tau}_k^B$, there exists a positive constant c_τ such that

$$|\tau_k| \leq c_\tau. \tag{4.8}$$

Write $\beta_{k+1}(\tau_k)$ as the special form of (1.6) with t replaced by p_k. It follows from (4.6), (4.7) and (4.8) that

$$|p_k| \leq \frac{L^2}{\mu} + L + c_\tau. \tag{4.9}$$

Consequently,

$$\begin{aligned} \|d_{k+1}\| &\leq \|g_{k+1}\| + \left| \frac{g_{k+1}^T y_k}{d_k^T y_k} - p_k \frac{g_{k+1}^T s_k}{d_k^T y_k} \right| \|d_k\| \\ &\leq \left(1 + \frac{L\,\|s_k\|\,\|d_k\|}{d_k^T y_k} + |p_k| \frac{\|s_k\|\,\|d_k\|}{d_k^T y_k} \right) \|g_{k+1}\| \\ &\leq \left(1 + \frac{L^2 + 2\mu L + \mu c_\tau}{\mu^2} \right) \|g_{k+1}\|. \end{aligned} \tag{4.10}$$

On the other hand, Lemmas 2.2 and 3.3 imply that

$$\sum_{k \geq 1} \frac{\|g_k\|^4}{\|d_k\|^2} < \infty. \tag{4.11}$$

By (4.10) and (4.11), we have that

$$\sum_{k \geq 1} \|g_k\|^2 < \infty,$$

which implies (4.5). □

Denote θ_k to be the angle between d_k and $-g_k$; namely,

$$\cos\theta_k = \frac{-g_k^T d_k}{\|g_k\|\,\|d_k\|}.$$

In the case that f is uniformly convex, we know from (2.23) and (4.10) that there must some positive constant c_θ such that

$$\cos\theta_k \geq c_\theta, \quad \forall\ k \geq 1.$$

For general nonlinear functions, similarly to [15] and [7], we can establish a weaker convergence result in the sense that

$$\liminf_{k \to \infty} \|g_k\| = 0. \tag{4.12}$$

To this aim, we proceed by contradiction and assuming that there exists $\gamma > 0$ such that

$$\|g_k\| \geq \gamma, \quad \forall\ k \geq 1. \tag{4.13}$$

Lemma 4.3. *Assume that f satisfies Assumption 1.1. Consider the family of conjugate gradient methods of the form* (1.3)*, where d_{k+1} is given by* (2.11) *and* (2.28) *and stepsize α_k is calculated by the improved Wolfe line search satisfying* (3.6) *and* (3.4)*. If* (4.13) *holds, then $d_k \neq 0$ and*

$$\sum_{k \geq 2} \|u_k - u_{k-1}\|^2 < \infty, \tag{4.14}$$

where $u_k = d_k / \|d_k\|$.

Proof. First, note that $d_k \neq 0$, for otherwise the sufficient descent condition (2.30) would imply $g_k = 0$. Therefore u_k is well defined. Now, divide formula (2.28) for β_k into two parts as follows

$$\begin{aligned} \beta_k^{(1)} &= \max\left\{ \frac{g_{k+1}^T y_k}{d_k^T y_k} - \left(1 + \tau_k \frac{s_k^T y_k}{\|y_k\|^2}\right) \frac{\|y_k\|^2}{d_k^T y_k} \frac{g_{k+1}^T d_k}{d_k^T y_k} + (1-\eta)\, \frac{g_{k+1}^T d_k}{\|d_k\|^2},\, 0 \right\}, \quad (4.15) \\ \beta_k^{(2)} &= \eta\, \frac{g_{k+1}^T d_k}{\|d_k\|^2} \quad (4.16) \end{aligned}$$

and define

$$w_k = \frac{-g_k + \beta_{k-1}^{(2)} d_{k-1}}{\|d_k\|} \quad \text{and} \quad \delta_k = \frac{\beta_{k-1}^{(1)} \|d_{k-1}\|}{\|d_k\|}. \tag{4.17}$$

By $d_k = -g_k + \beta_{k-1} d_{k-1}$, we have for $k \geq 2$,

$$u_k = w_k + \delta_k u_{k-1}. \tag{4.18}$$

Using the identity $\|u_k\| = \|u_{k-1}\| = 1$ and (4.18), we obtain

$$\|w_k\| = \|u_k - \delta_k u_{k-1}\| = \|\delta_k u_k - u_{k-1}\| \tag{4.19}$$

(the last equality can be verified by squaring both sides). Using the condition $\delta_k \geq 0$, the triangle inequality, and (4.19), we have

$$\begin{aligned} \|u_k - u_{k-1}\| &\leq \|(1+\delta_k)u_k - (1+\delta_k)u_{k-1}\| \\ &\leq \|u_k - \delta_k u_{k-1}\| + \|\delta_k u_k - u_{k-1}\| \\ &= 2\|w_k\|. \end{aligned} \tag{4.20}$$

By the definition of $\beta_k^{(2)}$ in (4.16), we see that

$$\| - g_k + \beta_{k-1}^{(2)} d_{k-1}\| \leq \|g_k\| + |\beta_{k-1}^{(2)}| \|d_{k-1}\| \leq (1+\eta)\|g_k\|. \tag{4.21}$$

This bound for the numerator of w_k coupled with (4.20) gives

$$\|u_k - u_{k-1}\| \leq 2\|w_k\| \leq 2(1+\eta)\, \frac{\|g_k\|}{\|d_k\|}. \tag{4.22}$$

The relation (4.13), the sufficient descent condition (2.30) and the Zoutendijk condition (3.11) indicate that

$$\sum_{k \geq 1} \frac{\|g_k\|^2}{\|d_k\|^2} \leq \frac{1}{\gamma^2} \sum_{k \geq 1} \frac{\|g_k\|^4}{\|d_k\|^2} \leq \frac{1}{\gamma^2\, \bar{c}^2} \sum_{k \geq 1} \frac{(g_k^T d_k)^2}{\|d_k\|^2} < +\infty. \tag{4.23}$$

Thus (4.14) follows from (4.22) and (4.23). □

Now we give the following convergence theorem for general objective functions.

Theorem 4.4. *Assume that f satisfies Assumption 1.1. Consider the family of methods of the form* (1.3)*, where d_{k+1} is given by* (2.11) *and* (2.28) *and stepsize α_k is calculated by the improved Wolfe line search satisfying* (3.6) *and* (3.4)*. If the generated sequence $\{x_k\}$ is bounded, if τ_k is chosen as any of τ_k^H, τ_k^B, $\bar{\tau}_k^H$ and $\bar{\tau}_k^B$, the method converges in the sense that* (4.12) *holds.*

Proof. We proceed by contradiction and assume that (4.13) holds. By the continuity of ∇f and the boundedness of $\{x_k\}$, there exists some positive constant $\bar{\gamma}$ such that

$$\|x_k\| \le \bar{\gamma}, \ \|g_k\| \le \bar{\gamma}, \quad \forall\ k \ge 1. \tag{4.24}$$

The line search condition (3.4) indicates that

$$g_{k+1}^T d_k \ge \sigma\, g_k^T d_k. \tag{4.25}$$

It follows from this, (2.30) and (4.13) that

$$d_k^T y_k \ge -(1-\sigma) d_k^T g_k \ge \bar{c}(1-\sigma)\gamma^2. \tag{4.26}$$

Also we have by (4.25) and $g_k^T d_k < 0$ that

$$\frac{\sigma}{\sigma - 1} \le \frac{d_k^T g_{k+1}}{d_k^T y_k} \le 1. \tag{4.27}$$

For any τ_k of τ_k^H, τ_k^B, $\bar{\tau}_k^H$ and $\bar{\tau}_k^B$, it is not difficult to know by (1.2) and (4.8) that there exists some positive constant $\bar{c}_\tau$ such that

$$\left|\tau_k\, s_k^T y_k\right| \le \bar{c}_\tau\, \|s_k\|^2, \quad \forall\ k \ge 1. \tag{4.28}$$

Now we write $\beta_k(\tau_k)$ in (2.12) as

$$\beta_k(\tau_k) = \frac{g_{k+1}^T y_k}{d_k^T y_k} - \left(1 - \frac{(d_k^T y_k)^2}{\|d_k\|^2\,\|y_k\|^2}\right) \frac{\|y_k\|^2}{d_k^T y_k} \frac{g_{k+1}^T d_k}{d_k^T y_k} - \frac{\tau_k\, s_k^T y_k}{d_k^T y_k} \frac{g_{k+1}^T d_k}{d_k^T y_k}. \tag{4.29}$$

Since by (4.24), $\|s_k\| = \|x_{k+1} - x_k\| \le 2\bar{\gamma}$, we can show by this, (4.29), (4.24), (4.26), (4.28), $\|y_k\| \le L\|s_k\|$ and $0 \le (d_k^T y_k)^2 \le \|d_k\|^2 \|y_k\|^2$ that

$$|\beta_k(\tau_k)| \le c_\beta \|s_k\|, \quad \text{for some constant } c_\beta > 0 \text{ and all } k \ge 1. \tag{4.30}$$

Define $b = 2c_\beta\bar{\gamma}$ and $\lambda = \frac{1}{2c_\beta^2\bar{\gamma}}$. It follows from (4.30) and (4.24) that for all k,

$$|\beta_k| \le b, \tag{4.31}$$

and

$$\|s_k\| \le \lambda \implies |\beta_k| \le \frac{1}{b}. \tag{4.32}$$

The relations (4.31) and (4.32) indicate that $\beta_k(\tau_k)$ in (2.12) has Property $(*)$ in [15].

Now we look at the formula (2.28). By (4.13), (2.30) and (3.11), we clearly have that

$$\|d_k\| \to +\infty. \tag{4.33}$$

This means that $\beta_k(\tau_k)$ can only be less than the value $\beta_k^{(2)} = \eta \frac{g_{k+1}^T d_k}{\|d_k\|^2}$ for finite times. Otherwise, we have that

$$\|d_{k+1}\| = \|g_{k+1} + \beta_k^{(2)} d_k\| \le (1+\eta)\|g_{k+1}\| \le (1+\eta)\bar{\gamma}$$

for infinite k's and obtain a contradiction to (4.33). Consequently, we can assume that $\beta_k^+(\tau_k) = \beta_k(\tau_k)$ for all sufficiently large k. In this case, using Property $(*)$ and the fact that $\|d_k\|^2$ is increasing at most linearly, we can show similarly to Lemma 4.2 in [15] that for any positive integers Δ and k_0, there exists an integer $k \ge k_0$ such that the size of $\mathcal{K} = \{i : k \le i \le k + \Delta - 1, \|s_{i-1}\| > \lambda\}$ is greater than $\frac{\Delta}{2}$. Further, using this, Lemma 4.3 and the boundedness of $\{x_k\}$, we can obtain a contradiction similarly to the proof of Theorem 4.3 in [15]. The contradiction shows the truth of (4.12). □

5 Numerical Experiments

In this section, we compare Algorithm 4.1 with the cg_descent method (C version 3.0) of Hager and Zhang in [17]. We also examine the performance of new conjugate gradient variants introduced in Section 2 using the same line search in the cg_descent method. Our numerical experiments indicate that the scheme in the Hager-Zhang family corresponding to $\theta_k \equiv 1$ (that is (1.15)) performs better than both the $\theta_k \equiv 2$ scheme (that is (1.8)) and all the other schemes that were tested.

Algorithm 4.1 was implemented by modifying the Hager-Zhang cg_descent C code, version 3.0, to incorporate the improved Wolfe line search, adaptive restarts, and the flexibility to test different values for β_k. We adjusted some of the parameter values in cg_descent in order to improve the performance. The following parameters were used in our implementation.

$$\varepsilon = 10^{-6},\ \delta = 0.1,\ \sigma = 0.9,\ \psi = 5,\ \epsilon_1 = 10^{-3},\ \epsilon_2 = 100,\ \epsilon_3 = 10^{-10},$$
$$\epsilon_4 = 10^{-3},\ \rho = 5,\ \eta = 0.5,\ M = 10^{10},\ MaxRestart = 6\,n,\ MinQuad = 3,$$

where n is the problem dimension. The computer used is a Lenovo X200 laptop with 2G RAM memory and Centrino2 processor. The initial stepsize $\alpha_1^{(0)}$ at the first iteration is chosen in the same way as in [18]. We used the same termination criterion as in cg_descent method [17]; namely,

$$\|\nabla f(x_k)\|_\infty \leq 10^{-6}. \tag{5.1}$$

We adopted the same basic collection of CUTEr [16] unconstrained test problems as in [18], which can be found in the website http://www.math.ufl.edu/˜hager/papers/CG/ACM.stats. There are 118 test problems altogether and their dimensions vary from 50 and 10^4. For each comparison, however, we excluded those problems for which different solvers converge to different local minimizers.

The performance profile by Dolan and Morè [11] is used to display the performance of the algorithms. Define $\mathcal{P}$ as the whole set of n_p test problems and $\mathcal{S}$ the set of the interested solvers. Denote $nf_{p,s}$ and $ng_{g,s}$ to be the number of objective function evaluations and the number of gradient evaluations, respectively, required by solver s for problem p. Let $l_{p,s} = nf_{p,s} + 3\,ng_{p,s}$ (this multiplier 3 is reasonable due to automatic differentiation theory and a larger multiplier will favor our algorithm since it often uses fewer gradient evaluations) and define the performance ratio as

$$r_{p,s} = \frac{l_{p,s}}{l_p^*},$$

where $l_p^* = \min\{l_{p,s} : s \in \mathcal{S}\}$. It is obvious that $r_{p,s} \geq 1$ for all p and s. If a solver fails to solve a problem, the ratio $r_{p,s}$ is assigned to be a large number 10^{10}. The performance profile for each solver s is defined as the following cumulative distribution function for performance ratio $r_{p,s}$,

$$\rho_s(\tau) = \frac{size\{p \in \mathcal{P} : r_{p,s} \leq \tau\}}{n_p}.$$

Obviously, $\rho_s(1)$ represents the percentage of problems for which solver s is the best. See [11] for more details about the performance profile. The performance profile can also be used to analyze the cpu time.

Figure 1 plots the performance profile with the four variants of the new algorithms; namely, Algorithms 4.1 (a), 4.1 (b), 4.1 (c) and 4.1 (d). After eliminating the problems for which the four variants converge to different local minimizers, 105 problems are left. Observe that among the four algorithms, Algorithm 4.1 (b) occupies the first place, which is fastest for about 50% of the test problems; Algorithms 4.1 (c) and 4.1 (d) come second and Algorithm 4.1 (a) third. It indicates that the new choice β_k (2.28) with τ_k^B in (2.7) is more efficient than the one with τ_k^H in (2.6). Note that this top performing choice for β_k is the member of the Hager-Zhang family corresponding to $\theta_k \equiv 1$.

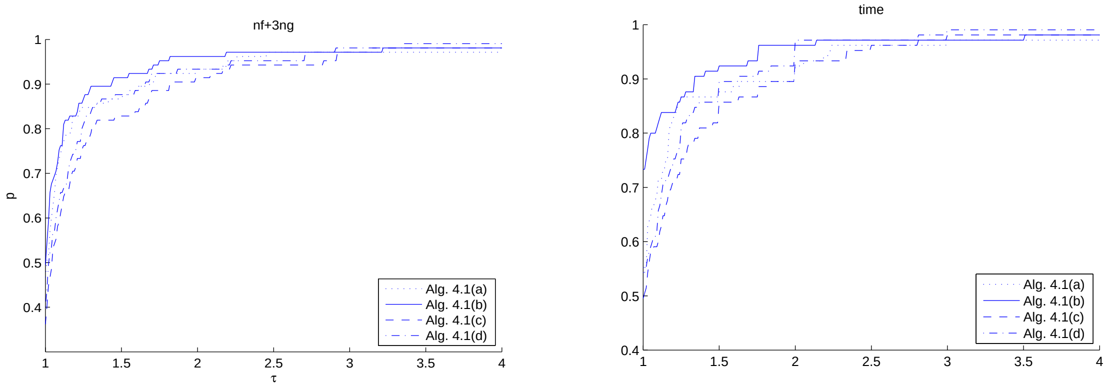

Figure 1: Performance profile of Algorithms 4.1 (a), (b), (c) and (d) based on the numbers of function/gradient evaluations $l_{p,s}$ (left) and CPU time (right).

In Figure 2, we compare Algorithm 4.1 (b) with and without adaptive restarts to test the efficiency of the new restart technique. Here a regular restart strategy of setting $d_k = -g_k$ once $k - k_l \geq 6n$ is used in both cases, where k_l is the last restart iteration. The difference is whether the algorithm is restarted or not based on how the quantity r_k tends to 1 or equivalently how the function is close to be quadratic approximately (see Step 3 of Algorithm 4.1 for details). There are 115 problems left after the elimination process mentioned above. Figure 2 shows that the adaptive restart technique contributes to the efficiency of Algorithm 4.1 (b). Similar observations were also made for Algorithms 4.1 (a), (c) and (d).

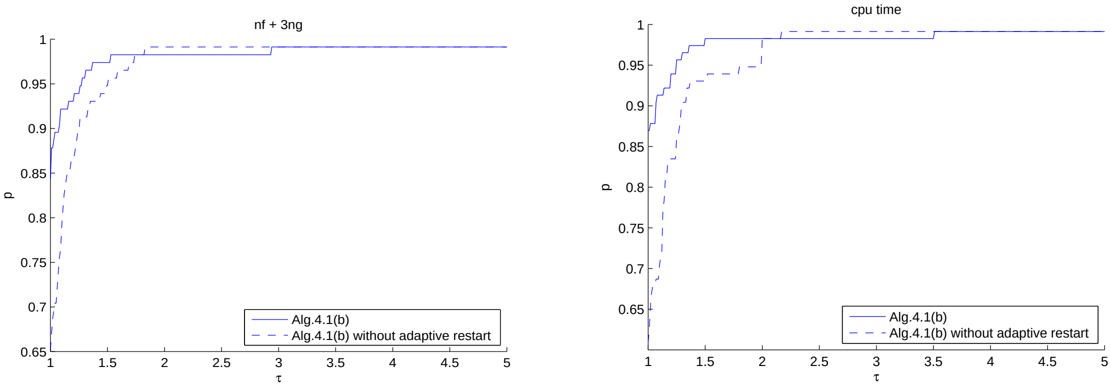

Figure 2: Performance profile of Algorithms 4.1 with and without adaptive restarts based on the numbers of function/gradient evaluations $l_{p,s}$ (left) and CPU time (right).

In Figure 3, we compare cg_descent itself (version 3.0) and its variant with β_k replaced by the choice of β_k given by (2.12) and (2.7) which is corresponding to $\theta_k \equiv 1$ in (1.11). There are 107 problems left after the elimination process mentioned above. From Figure 3, we see that the $\theta_k \equiv 1$ in Hager-Zhang family performs better than $\theta_k \equiv 2$.

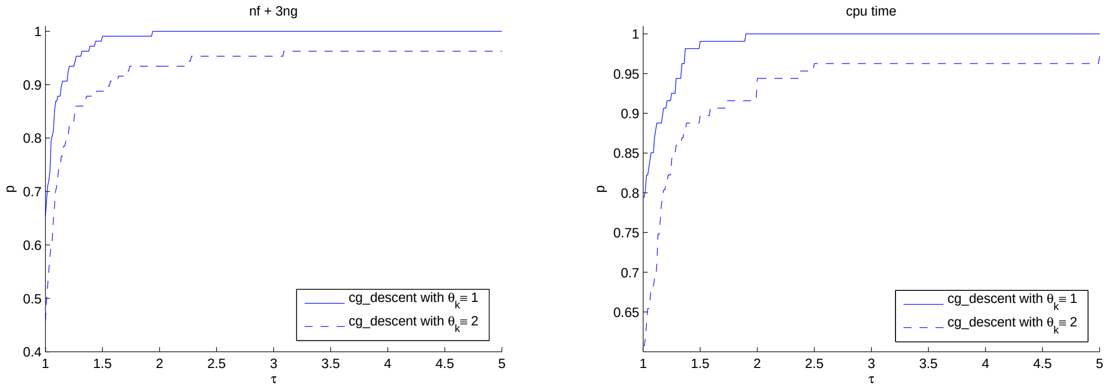

Figure 3: Performance profile of cg_descent with $\theta_k \equiv 1$ and $\theta_k \equiv 2$ based on the number of function/gradient evaluations $l_{p,s}$ (left) and CPU time (right).

As requested by one referee, we also compared Algorithm 4.1(b) with cg_descnet 3.0 and its newest version (version 5.3) for the entire test set (see Figure 4). In version 5.3 of cg_descent, the parameter β_k has been calculated by (1.15) with a similar truncation as in (2.28). A restart procedure similar to the proposal in this paper has also been used in this version. Therefore by this comparison, we mainly evaluate the relative performance of the improved Wolfe line search described in Section 3. There are 104 problems left after the elimination process mentioned above. From Figure 4 we can see that Algorithm 4.1(b) has a significant improvement over cg_descent (version 3.0). Compared with version 5.3 of cg_descent, Algorithm 4.1(b) favors more function evaluations, but less gradient evaluations. For non-convex functions, as shown by the results of Section 4, Algorithm 4.1 with the improved line search is globally convergent, whereas there is no guarantee for the global convergence of cg_descent. Figure 4 further shows the usefulness of the improved Wolfe line search in practical computations.

6 Conclusions and Discussions

We have proposed a family of conjugate gradient methods, namely, (1.3), (1.4) and (1.12), for unconstrained optimization via seeking the conjugate gradient direction closest to the direction of the scaled memoryless BFGS method. The sufficient descent condition is established for four special members of the family. An improved Wolfe line search has been introduced, which can avoid a numerical drawback of the Wolfe line search observed in [17] and guarantee the global convergence of the conjugate gradient method under mild conditions. Besides it, we have developed a new strategy to choose the initial stepsize and a dynamic restart technique to accelerate the algorithm. The numerical results indicate that Algorithm 4.1 (b), that calculates β_k by (2.28) and (2.7) or equivalently by (2.32), performs much better than the cg_descent method by Hager and Zhang [18] for the test problems from the CUTEr collection.

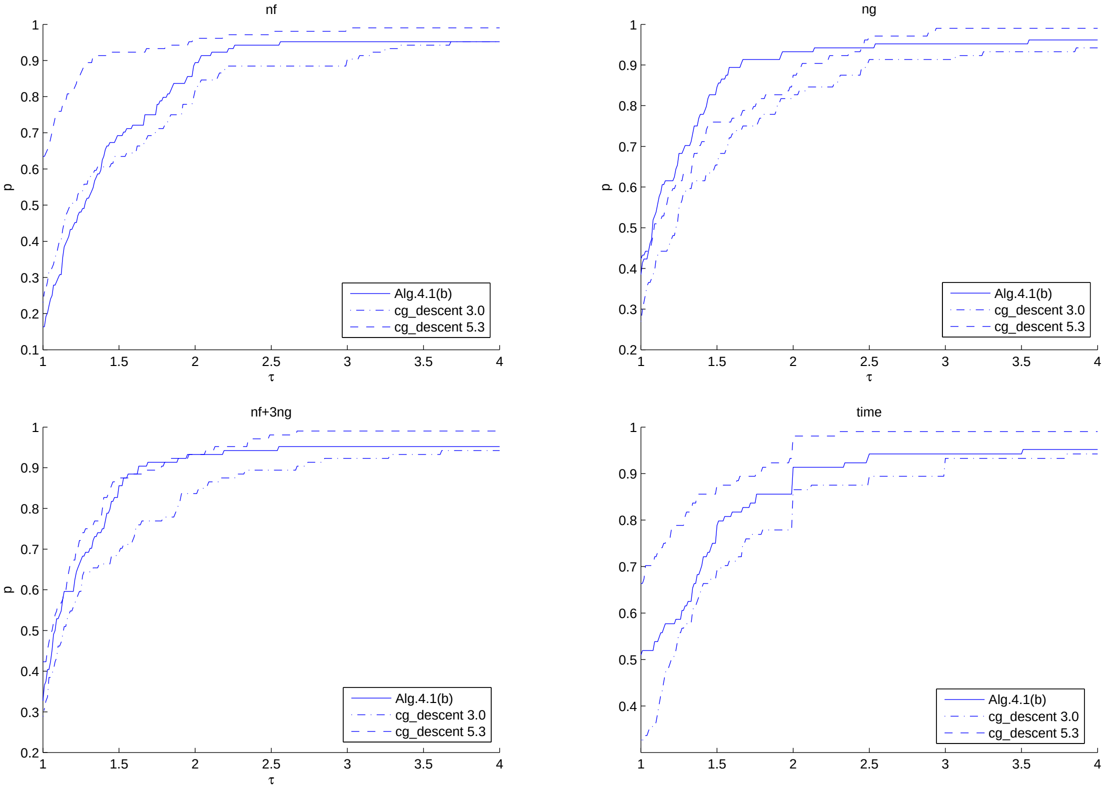

Figure 4: Performance profile of Algorithms 4.1(b), cg_descent 3.0 and 5.3 based on the number of function evaluations/gradient evaluations and CPU time.

To a great extent, both the new family (1.12) proposed in this paper and the Hager-Zhang family (1.11) could be regarded as subfamilies of the Dai-Liao family (1.6). Comparing the new family with the Hager-Zhang one, the parameter τ_k in (1.12) has a clear meaning; namely, it is corresponding to the self-scaling parameter in the scaled memoryless BFGS method. On the occasion of quasi-Newton methods, to improve the condition numbers of quasi-Newton matrices, this parameter τ_k (see [25, 26]) must be such that

$$\tau_k^B \leq \tau_k \leq \tau_k^H, \tag{6.1}$$

where τ_k^B and τ_k^H are given in (2.7) and (2.6), respectively. This suggested the following interval of the quantity t_k in (2.13),

$$t_k \in \left[\frac{\|y_k\|^2}{s_k^T y_k}, 2\frac{\|y_k\|^2}{s_k^T y_k} - \frac{s_k^T y_k}{\|s_k\|^2} \right]. \tag{6.2}$$

In this case, the new family of methods (1.12) does not include the formula (1.8) by Hager and Zhang [17] if $s_k^T y_k > 0$ for all k. We wonder whether this suggested interval (6.2) of t_k is helpful in nonlinear conjugate gradient field. In addition, since many choices on the self-scaling parameter τ_k have been proposed in [1, 25, 26, 27] and the references therein, we wonder if there exist any other members of the new family (1.12) which are more efficient than (2.28). This still remains under investigation.

As seen from Figure 2, the proposed dynamic restart stratgy indeed contributes to the efficiency of Algorithm 4.1 (b). This is mainly based on the quantity r_{k-1} defined by (4.3), which reflects how the function is close to some quadratic function in some sense. Another useful quantity in designing dynamical restart strategy is

$$\xi_{k-1} = \frac{g_k^T g_{k-1}}{\|g_k\|^2}. \tag{6.3}$$

Specifically, Powell [31] introduced the restart criterion $|\xi_{k-1}| \geq 0.2$ for Beale [2]'s three-term conjugate gradient method and obtained satisfactory numerical results. Such a restart criterion was also used by Shanno and his collaborator [34, 36] in building the CONMIN software. Although our numerical results showed that the naive replacement of the adaptive restart criterion in Step 3 of Algorithm 4.1(b) by Powell's restart criterion is not so good, we feel that there is still a broad room how to develop more efficient restart strategy in the design of nonlinear conjugate gradient algorithms.

To extend the idea of this paper, we may consider the self-scaling memoryless Broyden family of methods, whose search direction is parallel to

$$\begin{aligned} d_{k+1} &= -g_{k+1} + \left[\theta_k \frac{y_k^T y_k}{s_k^T y_k} \left(\frac{g_{k+1}^T y_k}{y_k^T y_k} - \frac{g_{k+1}^T s_k}{s_k^T y_k} \right) - \tau_k \frac{g_{k+1}^T s_k}{s_k^T y_k} \right] s_k \\ &\quad + \left[\frac{g_{k+1}^T y_k}{y_k^T y_k} + \theta_k \left(\frac{g_{k+1}^T s_k}{s_k^T y_k} - \frac{g_{k+1}^T y_k}{y_k^T y_k} \right) \right] y_k, \end{aligned} \tag{6.4}$$

where τ_k is the scaling parameter again and θ_k is the parameter related to the Broyden's family. By projecting the above direction into the one-dimensional manifold $\mathcal{S}$ in (2.9), we can obtain the two-parameter family of methods where $d_{k+1} = -g_{k+1} + \beta_k(\tau_k, \theta_k)\, d_k$ and

$$\beta_k(\tau_k, \theta_k) = \left[\theta_k \left(\frac{y_k^T y_k}{d_k^T y_k} - \frac{d_k^T y_k}{\|d_k\|^2} \right) + \frac{d_k^T y_k}{\|d_k\|^2} \right] \frac{g_{k+1}^T y_k}{y_k^T y_k} - \left[\theta_k \left(\frac{y_k^T y_k}{s_k^T y_k} - \frac{s_k^T y_k}{s_k^T s_k} \right) + \tau_k \right] \frac{g_{k+1}^T s_k}{d_k^T y_k}. \tag{6.5}$$

If the line search is exact, in which case $g_{k+1}^T s_k = 0$, the above formula reduces to

$$\beta_k(\tau_k, \theta_k) = \left[\theta_k + (1-\theta_k)\frac{(d_k^T y_k)^2}{\|d_k\|^2\|y_k\|^2}\right]\frac{g_{k+1}^T y_k}{d_k^T y_k}. \tag{6.6}$$

Thus we can see that, the above two-parameter family of methods reduce to the linear conjugate gradient method only when $\theta_k = 1$, provided that the vectors d_k and y_k are not always parallel. Nevertheless, we might consider some dynamical ways of choosing θ_k. This remains under investigation.

Acknowledgements. The authors are very grateful to the anonymous referees for their useful suggestions and comments, which improved the quality of this paper.

References

[1] M. Al-Baali, *Numerical experience with a class of self-scaling quasi-Newton algorithms*, J. Optim. Theory and Appl., 96:3 (1998), pp. 533-553.

[2] E. M. L. Beale, *A derivation of conjugate gradients*, in Numerical Methods for Nonlinear Optimization, F. A. Lootsman, ed., Academic Press, London, 1972, pp. 39-43.

[3] W. Y. Cheng and Q. F. Liu, *Sufficient descent nonlinear conjugate gradient methods with conjugacy conditions*, Numerical Algorithms 53 (2010), pp. 113-131.

[4] Y. H. Dai, *A family of hybrid conjugate gradient methods for unconstrained optimization*, Mathematics of Computation, 72 (2003), pp. 1317-1328.

[5] Y. H. Dai, *Nonlinear Conjugate Gradient Methods*, Wiley Encyclopedia of Operations Research and Management SciencePublished OnlineFeb 2011, DOI: 10.1002/9780470400531.eorms0183/pdf

[6] Y.H. Dai, J. Han, G. Liu, D. Sun, H. Yin, and Y. Yuan, *Convergence properties of nonlinear conjugate gradient methods*, SIAM J. Optim., 10 (1999), pp. 345C358.

[7] Y. H. Dai and L. Z. Liao, *New conjugacy conditions and related nonlinear conjugate gradient methods*, Appl. Math. Optim., 43 (2001), pp. 87-101.

[8] Y. H. Dai and Y. Yuan, *A nonlinear conjugate gradient method with a strong global convergence property*, SIAM J. Optim., 10 (1999), pp. 177-182.

[9] Y. H. Dai and Y. Yuan, *An efficient hybrid conjugate gradient method for unconstrained optimization*, Annals of Operations Research, 103 (2001), pp. 33-47.

[10] Y. H. Dai and H. Zhang, *An adaptive two-point stepsize gradient algorithm*, Numerical Algorithms, 27 (2001), pp. 377-385.

[11] E. D. Dolan and J. J. Moré, *Benchmarking optimization software with performance profiles*, Math. Programming, 91 (2002), pp. 201-213.

[12] R. Fletcher, *A FORTRAN subroutine for minimization by the method of conjugate gradients*, Report R7073, U. K. A. E. R. E., Harwell, England, 1972.

[13] R. Fletcher, *Practical Methods of Optimization vol. 1: Unconstrained Optimization*, John Wiley & Sons, New York, 1987.

[14] R. Fletcher and C. Reeves, *Function minimization by conjugate gradients*, Comput. J., 7 (1964), pp. 149-154.

[15] J. C. Gilbert and J. Nocedal, *Global convergence properties of conjugate gradient methods for optimization*, SIAM J. Optim., 2:1 (1992), pp. 21-42.

[16] N. I. M. Gould, D. Orban and Ph. L. Toint, *CUTEr (and SifDec), a constrained and unconstrained testing environment, revisited*, Technical Report TR/PA/01/04, CERFACS, Toulouse, France, 2001.

[17] W. W. Hager and H. Zhang, *A new conjugate gradient method with guaranteed descent and an efficient line search*, SIAM J. Optim., 16 (2005), pp. 170-192.

[18] W. W. Hager and H. Zhang, *Algorithm 851: CG_DESCENT, A conjugate gradient method with guaranteed descent*, ACM Transactions on Mathematical Software, 32 (2006), pp. 113-137.

[19] W. W. Hager and H. Zhang, *A survey of nonlinear conjugate gradient methods*, Pacific J. Optim., 2 (2006), pp. 35-58.

[20] M. R. Hestenes and E. L. Stiefel, *Methods of conjugate gradients for solving linear systems*, J. Research Nat. Bur. Standards, 49 (1952), pp. 409-436.

[21] G. Y. Li, C. M. Tang and Z. X. Wei, *New conjugacy condition and related new conjugate gradient methods for unconstrained optimization*, J. Comp. and Appl. Math., 202 (2007), pp. 523-539.

[22] J. J. Moré, B. S. Garbow and K. E. Hillstrom, *Testing unconstrained optimization software*, ACM Transactions on Mathematical Software, 7 (1981), pp. 17-41.

[23] J. Nocedal and S. J. Wright, *Numerical Optimization*, Springer Series in Opertions Research, Springer, New York, 1999.

[24] S. S. Oren, *Self-scaling variable metric algorithms for unconstrained minimization*, Ph.D. thesis, Department of Engineering Economic Systems, Stanford University, Stanford, Calif., 1972.

[25] S. S. Oren, *Self scaling variable metric (SSVM) algorithms, part II: Implementation and Experiments*, Management Science, 20:5 (1974), pp. 863-874.

[26] S. S. Oren and D. G. Luenberger, *Self scaling variable metric (SSVM) algorithms, part I: criteria and sufficient conditions for scaling a class of algorithms*, Management Science, 20:5 (1974), pp. 845-862.

[27] S. S. Oren and E. Spedicato, *Optimal conditioning of self scaling variable metric algorithms*, Math. Programming, 10:1 (1976), pp. 70-90.

[28] J. M. Perry, *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.

[29] E. Polak and G. Ribière, *Note sur la convergence de méthodes de directions conjugées*, Rev. Francaise Informat. Recherche Opértionelle, 3 (1969), pp. 35-43.

[30] B. T. Polyak, *The conjugate gradient method in extreme problems*, USSR Comp. Math. and Math. Phys., 9 (1969), pp. 94-112.

[31] M. J. D. Powell, *Restart procedures for the conjugate gradient method*, Math. Programming, 12 (1977), pp. 241-254.

[32] M. J. D. Powell, *Nonconvex minimization calculations and the conjugate gradient method.* in Lecture Notes in Mathematics 1066, D. F. Griffiths, ed., Springer, Berlin, 1984, pp. 122-141.

[33] M. J. D. Powell, *Convergence properties of algorithms for nonlinear optimization*, SIAM Review, 28 (1986), pp. 487-500.

[34] D. F. Shanno, *On the convergence of a new conjugate gradient algorithm*, SIAM J. Numer. Anal., 15 (1978), pp. 1247-1257.

[35] D. F. Shanno, *Remark on Algorithm 500*, ACM Transactions on Mathematical Software, 6 (1980), pp. 618-622.

[36] D. F. Shanno and K. H. Phua, *Matrix conditioning and nonlinear optimization*, Math. Programming, 14 (1978), pp. 149-160.

[37] P. Wolfe, *Convergence conditions for ascent methods*, SIAM Rev., 11 (1969), pp. 226-235.

[38] P. Wolfe, *Convengence conditions for ascent methods II: some corrections*, SIAM Rev., 13 (1971), pp. 185-188.

[39] H. Yabe and M. Takano, *Global convergence properties of nonlinear conjugate gradient methods with modified secant condition*, Comput. Optim. Appl., 28 (2004), pp. 203-225.

[40] G. H. Yu, L. T. Guan, *New descent nonlinear conjugate gradient methods for large-scale optimization*, Technical Report, Department of Scientific Computation and Computer Applications, Sun Yat-Sen University, Guangzhou, P. R. China, 2005.

[41] G. H. Yu, L. T. Guan and W. F. Chen, *Spectral conjugate gradient methods with sufficient descent property for large-scale unconstrained optimization*, Optimization Methods and Software 23:2 (2008), pp. 275-293.

[42] L. Zhang, W. Zhou and D. Li, *Global convergence of a modified Fletcher-Reeves conjugate gradient method with Armijo-type line search*, Numerische Mathematik 104:4 (2006), pp. 561 - 572.

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