J Optim Theory Appl (2009) 141: 249–264
DOI 10.1007/s10957-008-9505-0

Hybrid Conjugate Gradient Algorithm for Unconstrained Optimization

N. Andrei

Published online: 31 December 2008

Abstract In this paper a new hybrid conjugate gradient algorithm is proposed and analyzed. The parameter β_k is computed as a convex combination of the Polak-Ribière-Polyak and the Dai-Yuan conjugate gradient algorithms, i.e. $\beta_k^N = (1-\theta_k)\beta_k^{PRP} + \theta_k\beta_k^{DY}$. The parameter θ_k in the convex combination is computed in such a way that the conjugacy condition is satisfied, independently of the line search. The line search uses the standard Wolfe conditions. The algorithm generates descent directions and when the iterates jam the directions satisfy the sufficient descent condition. Numerical comparisons with conjugate gradient algorithms using a set of 750 unconstrained optimization problems, some of them from the CUTE library, show that this hybrid computational scheme outperforms the known hybrid conjugate gradient algorithms.

Keywords Unconstrained optimization · Hybrid conjugate gradient method · Conjugacy condition · Numerical comparisons

1 Introduction

Let us consider the nonlinear unconstrained optimization problem

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

where $f : R^n \to R$ is a continuously differentiable function, bounded from below. For solving this problem, starting from an initial guess $x_0 \in R^n$, a nonlinear conjugate

Communicated by F.A. Potra.

N. Andrei is a member of the Academy of Romanian Scientists, Splaiul Independenţei nr. 54, Sector 5, Bucharest, Romania.

N. Andrei (✉)
Research Institute for Informatics, Center for Advanced Modeling and Optimization, Averescu Avenue nr. 8-10, Sector 1, Bucharest, Romania
e-mail: nandrei@ici.ro

gradient method generates a sequence $\{x_k\}$ as

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

where $\alpha_k > 0$ is obtained by line search and the directions d_k are generated as

$$d_{k+1} = -g_{k+1} + \beta_k s_k, \quad d_0 = -g_0. \tag{3}$$

In (3), β_k is known as the conjugate gradient parameter, $s_k = x_{k+1} - x_k$ and $g_k = \nabla f(x_k)$. Consider $\|.\|$ the Euclidean norm and define $y_k = g_{k+1} - g_k$. The line search in the conjugate gradient algorithms is often based on the standard Wolfe conditions:

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

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

where d_k is a descent direction and $0 < \rho \leq \sigma < 1$. Plenty of conjugate gradient methods are known and an excellent survey of them, with special attention on their global convergence, is given by Hager and Zhang [1]. Different conjugate gradient algorithms correspond to different choices for the scalar parameter β_k. Some of these methods, such as Fletcher and Reeves (FR) [2], Dai and Yuan (DY) [3] and Conjugate Descent (CD) proposed by Fletcher [4],

$$\beta_k^{FR} = \frac{g_{k+1}^T g_{k+1}}{g_k^T g_k}, \qquad \beta_k^{DY} = \frac{g_{k+1}^T g_{k+1}}{y_k^T s_k}, \qquad \beta_k^{CD} = \frac{g_{k+1}^T g_{k+1}}{-g_k^T s_k},$$

have strong convergence properties, but they may have modest practical performance due to jamming. On the other hand, the methods of Polak–Ribière [5] and Polyak (PRP) [6], Hestenes and Stiefel (HS) [7] or Liu and Storey (LS) [8],

$$\beta_k^{PRP} = \frac{g_{k+1}^T y_k}{g_k^T g_k}, \qquad \beta_k^{HS} = \frac{g_{k+1}^T y_k}{y_k^T s_k}, \qquad \beta_k^{LS} = \frac{g_{k+1}^T y_k}{-g_k^T s_k},$$

may not generally be convergent, but they often have better computational performance.

In this paper we focus on hybrid conjugate gradient methods. These methods are combinations of different conjugate gradient algorithms. Their idea is to use the projections. They are mainly proposed in order to avoid the jamming phenomenon. One of the first hybrid conjugate gradient algorithms was introduced by Touati-Ahmed and Storey [9], where the parameter β_k is computed as

$$\beta_k^{TS} = \begin{cases} \beta_k^{PRP} = \frac{g_{k+1}^T y_k}{\|g_k\|^2}, & \text{if } 0 \leq \beta_k^{PRP} \leq \beta_k^{FR}, \\ \beta_k^{FR} = \frac{\|g_{k+1}\|^2}{\|g_k\|^2}, & \text{otherwise.} \end{cases}$$

The PRP method has a built-in restart feature that directly addresses the jamming problem. Indeed, when the step s_k is small, then the factor y_k in the numerator of

β_k^{PRP} tends to zero. Therefore, β_k^{PRP} becomes small and the search direction d_{k+1} is very close to the steepest descent direction $-g_{k+1}$. Hence, when the iterations jam, the method of Touati-Ahmed and Storey uses the PRP computational scheme.

Another hybrid conjugate gradient method was given by Hu and Storey [10], where β_k in (3) is

$$\beta_k^{HuS} = \max\{0, \min\{\beta_k^{PRP}, \beta_k^{FR}\}\}.$$

As above, when the method of Hu and Storey jams, then the PRP method is used instead.

The combination between LS and CD conjugate gradient methods leads to the following hybrid method:

$$\beta_k^{LS\text{-}CD} = \max\{0, \min\{\beta_k^{LS}, \beta_k^{CD}\}\}.$$

The CD method of Fletcher [4] is very close to the FR method. With an exact line search, the CD method is identical to FR. Similarly, for an exact line search, the LS method is also identical to PRP. Therefore, the hybrid LS-CD method with an exact line search has a similar performance with the hybrid method of Hu and Storey.

Gilbert and Nocedal in [11] suggested a combination between the PRP and the FR methods as:

$$\beta_k^{GN} = \max\{-\beta_k^{FR}, \min\{\beta_k^{PRP}, \beta_k^{FR}\}\}.$$

Since β_k^{FR} is always nonnegative, it follows that β_k^{GN} can be negative. The method of Gilbert and Nocedal has the same advantage of avoiding jamming.

Using the standard Wolfe line search, the DY method always generates descent directions and, if the gradient is Lipschitz continuous, the method is global convergent. In an effort to improve their algorithm, Dai and Yuan in [12] combined their algorithm with the Hestenes/Stiefel algorithm, proposing the following two hybrid methods:

$$\beta_k^{hDY} = \max\{-c\beta_k^{DY}, \min\{\beta_k^{HS}, \beta_k^{DY}\}\},$$

$$\beta_k^{hDYz} = \max\{0, \min\{\beta_k^{HS}, \beta_k^{DY}\}\},$$

where $c = (1-\sigma)/(1+\sigma)$. For the standard Wolfe conditions (4) and (5), under the Lipschitz continuity of the gradient, Dai and Yuan [12] established the global convergence of these hybrid computational schemes.

In this paper, we suggest another approach to get a hybrid conjugate gradient algorithm. Our hybrid algorithm is a convex combination of the PRP and DY conjugate gradient algorithms. We selected to combine these two methods in a hybrid conjugate gradient algorithm because PRP has good computational properties, on the one side, and DY has strong convergence properties, on the other side. Often the PRP method performs better in practice than DY and we make use of it in order to construct a good practical conjugate gradient algorithm. In Sect. 2, we introduce our hybrid conjugate gradient algorithm and prove that it generates descent directions satisfying under some conditions the sufficient descent condition. Section 3 presents our hybrid algorithm and Sect. 4 shows its convergence analysis. In Sect. 5, some numerical experiments and performance profiles of Dolan-Moré [13] corresponding

to this new hybrid conjugate gradient algorithm versus some other conjugate gradient algorithms are presented. The performance profiles corresponding to a set of 750 unconstrained optimization problems in the CUTE test problem library [14] as well as some other unconstrained optimization problems presented in [15] show that this hybrid conjugate gradient algorithm outperforms the known hybrid conjugate gradient algorithms.

2 Hybrid Conjugate Gradient Algorithm

The iterates $x_0, x_1, x_2, \ldots$ of our algorithm are computed by means of the recurrence (2), where the stepsize $\alpha_k > 0$ is determined according to the Wolfe conditions (4) and (5), the directions d_k are generated by the rule

$$d_{k+1} = -g_{k+1} + \beta_k^N s_k, \quad d_0 = -g_0, \tag{6}$$

where

$$\beta_k^N = (1-\theta_k)\beta_k^{PRP} + \theta_k \beta_k^{DY} = (1-\theta_k)\frac{g_{k+1}^T y_k}{g_k^T g_k} + \theta_k \frac{g_{k+1}^T g_{k+1}}{y_k^T s_k}, \tag{7}$$

and θ_k is a scalar parameter satisfying $0 \le \theta_k \le 1$, which will be determined in a specific way to be described later. Observe that if, $\theta_k = 0$, then $\beta_k^N = \beta_k^{PRP}$, and if $\theta_k = 1$, then $\beta_k^N = \beta_k^{DY}$. On the other hand, if $0 < \theta_k < 1$, then β_k^N is a convex combination of β_k^{PRP} and β_k^{DY}.

Referring to the PRP method, Polak and Ribière [5] proved that, when the function f is strongly convex and the line search is exact, then the PRP method is global convergent. In an effort to understand the behavior of the PRP method, Powell [16] showed that if the steplength $s_k = x_{k+1} - x_k$ approaches zero, the line search is exact, and the gradient $\nabla f(x)$ is Lipschitz continuous, then the PRP method is globally convergent. Additionally, assuming that the search direction is a descent one, Yuan [17] established the global convergence of the PRP method for strongly convex functions and a Wolfe line search. For general nonlinear functions, the convergence of the PRP method is uncertain. Powell [18] gave a three-dimensional example in which the function to be minimized is not strongly convex, showing that, even with an exact line search, the PRP method may not converge to a stationary point. Later on, Dai in [19] presented another example, this time with a strongly convex function for which the PRP method fails to generate a descent direction. Therefore, theoretically the convergence of the PRP method is limited to strongly convex functions. For general nonlinear functions the convergence of the PRP method is established under restrictive conditions (Lipschitz continuity, exact line search and the stepsize tends to zero). However, the numerical experiments presented, for example, by Gilbert and Nocedal [11] proved that the PRP method is one of the best conjugate gradient methods, and this is the main motivation for its use in (7).

On the other hand, the DY method always generates descent directions, and Dai in [20] established a remarkable property for the DY conjugate gradient algorithm, relating the descent directions to the sufficient descent condition. It is shown that,

if there exist constants γ_1 and γ_2 such that $\gamma_1 \leq \|g_k\| \leq \gamma_2$ for all k, then for any $p \in (0,1)$, there exists a constant $c > 0$ such that the sufficient descent condition $g_i^T d_i \leq -c\|g_i\|^2$ holds for at least $\lfloor pk \rfloor$ indices $i \in [0,k]$, where $\lfloor j \rfloor$ denotes the largest integer $\leq j$. Therefore, this property of DY is the main reason for the use of this method in (7).

In our algorithm, the parameter θ_k is selected in such a way that at every iteration the conjugacy condition is satisfied independently of the line search. From this condition, we call the corresponding algorithm CCOMB. Obviously,

$$d_{k+1} = -g_{k+1} + (1-\theta_k)\frac{y_k^T g_{k+1}}{g_k^T g_k} s_k + \theta_k \frac{g_{k+1}^T g_{k+1}}{y_k^T s_k} s_k. \tag{8}$$

Hence, from the conjugacy condition $y_k^T d_{k+1} = 0$, after some algebra, we get

$$\theta_k = \frac{(y_k^T g_{k+1})(y_k^T s_k) - (y_k^T g_{k+1})(g_k^T g_k)}{(y_k^T g_{k+1})(y_k^T s_k) - (g_{k+1}^T g_{k+1})(g_k^T g_k)}. \tag{9}$$

Theorem 2.1 *In the algorithm* (2), (8), (9), *assume that* α_k *is determined by the Wolfe line search* (4)–(5). *If* $0 < \theta_k < 1$, *then the direction* d_{k+1} *given by* (8) *is a descent direction*.

Proof Since $0 < \theta_k < 1$, from (8) we get

$$\begin{aligned} g_{k+1}^T d_{k+1} &= -\|g_{k+1}\|^2 + (1-\theta_k)\frac{y_k^T g_{k+1}}{g_k^T g_k} g_{k+1}^T s_k + \theta_k \frac{g_{k+1}^T g_{k+1}}{y_k^T s_k} g_{k+1}^T s_k \\ &\leq -\|g_{k+1}\|^2 + \frac{y_k^T g_{k+1}}{g_k^T g_k} g_{k+1}^T s_k + \frac{g_{k+1}^T g_{k+1}}{y_k^T s_k} g_{k+1}^T s_k \\ &= \left(-1 + \frac{g_{k+1}^T s_k}{y_k^T s_k}\right)\|g_{k+1}\|^2 + \frac{y_k^T g_{k+1}}{g_k^T g_k} g_{k+1}^T s_k \\ &= \frac{g_k^T s_k}{y_k^T s_k}\|g_{k+1}\|^2 + \frac{y_k^T g_{k+1}}{g_k^T g_k} g_{k+1}^T s_k. \end{aligned} \tag{10}$$

But, $y_k^T s_k > 0$ by (5) and, since $g_k^T s_k \leq 0$, it follows that $\frac{g_k^T s_k}{y_k^T s_k}\|g_{k+1}\|^2 \leq 0$. When the iterations are in progress or when they jam, y_k becomes tiny while $\|g_k\|$ is bounded away from zero. Consequently:

$$\left|\frac{g_k^T s_k}{y_k^T s_k}\right|\|g_{k+1}\|^2 \geq \frac{(g_{k+1}^T y_k)(g_{k+1}^T s_k)}{\|g_k\|^2}.$$

Therefore, from (10), it follows that $g_{k+1}^T d_{k+1} \leq 0$, i.e. the direction d_{k+1} is a descent one. □

Theorem 2.2 *If* $0 < \theta_k < 1$, *then the direction* d_{k+1} *given by* (8) *satisfies the sufficient descent condition*

$$g_{k+1}^T d_{k+1} \leq -\left(1 - \theta_k \frac{g_{k+1}^T s_k}{y_k^T s_k}\right) \|g_{k+1}\|^2. \tag{11}$$

Proof From (8), we have

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

Observe that y_k becomes tiny while $\|g_k\|$ is bounded away from zero. Consequently, the last term in the above inequality becomes negligible. Since $y_k^T s_k > 0$ by (5) and since $g_{k+1}^T s_k = y_k^T s_k + g_k^T s_k < y_k^T s_k$, then $y_k^T s_k / g_{k+1}^T s_k > 1$. But $0 < \theta_k < 1$; therefore, $\theta_k \leq y_k^T s_k / g_{k+1}^T s_k$, i.e.

$$g_{k+1}^T d_{k+1} \leq -\left(1 - \theta_k \frac{g_{k+1}^T s_k}{y_k^T s_k}\right) \|g_{k+1}\|^2 \leq 0. \qquad \square$$

Observe that the parameter θ_k given by (9) can be outside the interval $[0, 1]$. However, in order to have a real convex combination in (7) the following rule is used: if $\theta_k \leq 0$, then set $\theta_k = 0$ in (7), i.e. $\beta_k^N = \beta_k^{PRP}$; if $\theta_k \geq 1$, then take $\theta_k = 1$ in (7), i.e. $\beta_k^N = \beta_k^{DY}$. Therefore, under this rule for θ_k selection, the direction d_{k+1} in (8) combines the properties of the PRP and the DY algorithms in a convex way.

3 CCOMB Algorithm

Step 1. *Initialization.* Select $x_0 \in R^n$ and the parameters $0 < \rho \leq \sigma < 1$. Compute $f(x_0)$ and g_0. Consider $d_0 = -g_0$ and set $\alpha_0 = 1/\|g_0\|$.

Step 2. *Test for Continuation of Iterations*. If $\|g_k\|_\infty \leq 10^{-6}$, then stop.

Step 3. *Line Search*. Compute $\alpha_k > 0$ satisfying the Wolfe line search condition (4) and (5) and update the variables, $x_{k+1} = x_k + \alpha_k d_k$. Compute $f(x_{k+1})$, g_{k+1} and $s_k = x_{k+1} - x_k$, $y_k = g_{k+1} - g_k$.

Step 4. θ_k *Parameter Computation*. If $(y_k^T g_{k+1})(y_k^T s_k) - (g_{k+1}^T g_{k+1})(g_k^T g_k) = 0$, then set $\theta_k = 0$; otherwise, compute θ_k as in (9).

Step 5. β_k^N *Conjugate Gradient Parameter Computation*. If $0 < \theta_k < 1$, then compute β_k^N as in (7). If $\theta_k \geq 1$, then set $\beta_k^N = \beta_k^{DY}$. If $\theta_k \leq 0$, then set $\beta_k^N = \beta_k^{PRP}$.

Step 6. *Direction Computation*. Compute $d = -g_{k+1} + \beta_k^N s_k$. If the restart criterion of Powell

$$|g_{k+1}^T g_k| \geq 0.2\|g_{k+1}\|^2, \tag{12}$$

is satisfied, then set $d_{k+1} = -g_{k+1}$; otherwise, define $d_{k+1} = d$. Compute the initial guess $\alpha_k = \alpha_{k-1}\|d_{k-1}\|/\|d_k\|$, set $k = k+1$ and continue with Step 2.

It is well known that, if f is bounded along the direction d_k, then there exists a stepsize α_k satisfying the Wolfe line search conditions (4) and (5). In our algorithm, when the Powell restart condition is satisfied, i.e. $|g_{k+1}^T g_k| \geq 0.2\|g_{k+1}\|^2$, then we restart the algorithm with the negative gradient $-g_{k+1}$. More sophisticated reasons for restarting the algorithms have been proposed in the literature (see [21]), but we are interested in the performance of a conjugate gradient algorithm that uses this restart criterion associated to a direction satisfying the conjugacy condition when $0 < \theta_k < 1$. Under reasonable assumptions, conditions (4), (5) and (12) are sufficient to prove the global convergence of the algorithm.

The first trial of the steplength crucially affects the practical behavior of the algorithm. At every iteration $k \geq 1$ in the line search, the starting guess for the step α_k is computed as $\alpha_{k-1}\|d_{k-1}\|_2/\|d_k\|_2$. This selection was introduced for the first time by Shanno and Phua in CONMIN [22]. It is also used in the packages SCG by Birgin and Martínez [23] and SCALCG by Andrei [24–26].

4 Convergence Analysis

Throughout this section, we assume that:

(i) *The level set* $S = \{x \in R^n : f(x) \leq f(x_0)\}$ *is bounded.*
(ii) *In a neighborhood* N *of* S, *the function* f *is continuously differentiable and its gradient is Lipschitz continuous, i.e. there exists a constant* $L > 0$ *such that* $\|\nabla f(x) - \nabla f(y)\| \leq L\|x - y\|$, *for all* $x, y \in N$.

Under these assumptions on f, there exists a constant $\Gamma \geq 0$ such that $\|\nabla f(x)\| \leq \Gamma$, for all $x \in S$. The convergence of the steepest descent method with Armijo-type search is proved under very general conditions in [27]. On the other hand, in [28] it is proved that, for any conjugate gradient method with strong Wolfe line search, the following general result holds.

Lemma 4.1 *Let assumptions* (i) *and* (ii) *hold and consider any conjugate gradient method* (2) *and* (3), *where* d_k *is a descent direction and* α_k *is obtained by the strong Wolfe line search. If*

$$\sum_{k\geq 1} \frac{1}{\|d_k\|^2} = \infty, \tag{13}$$

then

$$\liminf_{k\to\infty} \|g_k\| = 0. \tag{14}$$

For uniformly convex functions which satisfy the above assumptions, we can prove that the norm of d_{k+1} given by (8) is bounded above. Assume that the function

f is a uniformly convex function, i.e. there exists a constant $\mu \geq 0$ such that, for all $x, y \in S$,

$$(\nabla f(x) - \nabla f(y))^T (x - y) \geq \mu \|x - y\|^2, \tag{15}$$

and the steplength α_k is obtained by the strong Wolfe line search.

$$f(x_k + \alpha_k d_k) - f(x_k) \leq \rho \alpha_k g_k^T d_k, \tag{16}$$

$$|g_{k+1}^T d_k| \leq -\sigma g_k^T d_k. \tag{17}$$

Using Lemma 4.1 the following result can be proved.

Theorem 4.1 *Suppose that the assumptions* (i) *and* (ii) *hold. Consider the algorithm* (2), (8) *and* (9), *where* $0 \leq \theta_k \leq 1$ *and* α_k *is obtained by the strong Wolfe line search. If* $\|s_k\|$ *tends to zero and there exists nonnegative constants* η_1 *and* η_2 *such that*

$$\|g_k\|^2 \geq \eta_1 \|s_k\|^2, \qquad \|g_{k+1}\|^2 \leq \eta_2 \|s_k\|, \tag{18}$$

and f *is a uniformly convex function, then*

$$\lim_{k \to \infty} g_k = 0. \tag{19}$$

Proof From (15) it follows that $y_k^T s_k \geq \mu \|s_k\|^2$. Now, since $0 \leq \theta_k \leq 1$, from uniform convexity and (18) we have

$$|\beta_k^N| \leq \left| \frac{g_{k+1}^T y_k}{g_k^T g_k} \right| + \left| \frac{g_{k+1}^T g_{k+1}}{y_k^T s_k} \right| \leq \frac{\|g_{k+1}\| \|y_k\|}{\eta_1 \|s_k\|^2} + \frac{\eta_2 \|s_k\|}{\mu \|s_k\|^2}.$$

But $\|y_k\| \leq L \|s_k\|$,

$$|\beta_k^N| \leq \frac{\Gamma L}{\eta_1 \|s_k\|} + \frac{\eta_2}{\mu \|s_k\|}.$$

Hence,

$$\|d_{k+1}\| \leq \|g_{k+1}\| + |\beta_k^N| \|s_k\| \leq \Gamma + \frac{\Gamma L}{\eta_1} + \frac{\eta_2}{\mu},$$

which implies that (13) is true. Therefore, by Lemma 4.1 we have (14), which for uniformly convex functions is equivalent to (19). □

Powell [16] showed that, for general functions, the PRP method is globally convergent if the steplength $\|s_k\| = \|x_{k+1} - x_k\|$ tends to zero, i.e. $\|s_k\| \leq \|s_{k-1}\|$ is a condition of convergence. For the convergence of our algorithm from (17), we see that, for $k \geq 1$, the gradient must be bounded as: $\eta_1 \|s_k\|^2 \leq \|g_k\|^2 \leq \eta_2 \|s_{k-1}\|$. If the Powell condition is satisfied, i.e. $\|s_k\|$ tends to zero, then $\|s_k\|^2 \ll \|s_{k-1}\|$ and therefore the norm of the gradient can satisfy (18). In the numerical experiments, we observed that (18) is constantly satisfied in the last part of the iterations.

For general nonlinear functions, the convergence analysis of our algorithm exploits insights developed by Gilbert and Nocedal [11], by Dai and Liao [29] and by Hager and Zhang [30]. The global convergence proof of the CCOMB algorithm is based on the Zoutendijk condition combined with the analysis showing that the sufficient descent condition holds and $\|d_k\|$ is bounded. Suppose that the level set L is bounded and the function f is bounded from below.

Lemma 4.2 *Assume that d_k is a descent direction and ∇f satisfies the Lipschitz condition $\|\nabla f(x) - \nabla f(x_k)\| \le L\|x - x_k\|$ for all x on the line segment connecting x_k and x_{k+1}, where L is a constant. If the line search satisfies the second Wolfe condition (5), then*

$$\alpha_k \ge \frac{1-\sigma}{L}\frac{|g_k^T d_k|}{\|d_k\|^2}. \tag{20}$$

Proof Subtracting $g_k^T d_k$ from both sides of (5) and using the Lipschitz condition we have

$$(\sigma - 1)g_k^T d_k \le (g_{k+1} - g_k)^T d_k \le L\alpha_k \|d_k\|^2. \tag{21}$$

Since d_k is a descent direction and $\sigma < 1$, then (20) follows immediately from (21). □

Theorem 4.2 *Let assumptions* (i) *and* (ii) *hold. Assume that $0 < \theta_k < 1$ and that, for every $k \ge 0$, there exists a positive constant ω such that $1 - \theta_k(g_{k+1}^T s_k)/(y_k^T s_k) \ge \omega > 0$ as well as the constants γ and Γ such that $\gamma \le \|g_k\| \le \Gamma$. Then, for the computational scheme (2), (8), (9), where α_k is determined by the Wolfe line search (4) and (5), either $g_k = 0$ for some k or*

$$\liminf_{k\to\infty} \|g_k\| = 0. \tag{22}$$

Proof From the Wolfe condition (5) we have

$$y_k^T s_k = (g_{k+1} - g_k)^T s_k \ge (\sigma - 1)g_k^T s_k = -(1-\sigma)g_k^T s_k. \tag{23}$$

By Theorem 2.2 and the assumption $1 - \theta_k(g_{k+1}^T s_k)/(y_k^T s_k) \ge \omega$, it follows that

$$g_k^T d_k \le -\left(1 - \theta_{k-1}\frac{g_k^T s_{k-1}}{y_{k-1}^T s_{k-1}}\right)\|g_k\|^2 \le -\omega\|g_k\|^2.$$

Therefore,

$$-g_k^T d_k \ge \omega\|g_k\|^2. \tag{24}$$

Combining (23) with (24), we get

$$y_k^T s_k \ge (1-\sigma)\omega\alpha_k\gamma^2.$$

On the other hand, $\|y_k\| = \|g_{k+1} - g_k\| \le L\|s_k\|$. Hence,

$$|g_{k+1}^T y_k| \le \|g_{k+1}\|\|y_k\| \le \Gamma L\|s_k\|.$$

With these, from (7) we get

$$|\beta_k^N| \leq \left|\frac{g_{k+1}^T y_k}{g_k^T g_k}\right| + \left|\frac{g_{k+1}^T g_{k+1}}{y_k^T s_k}\right|.$$

But,

$$\left|\frac{g_{k+1}^T y_k}{g_k^T g_k}\right| \leq \frac{\|g_{k+1}\|\|y_k\|}{\gamma^2} \leq \frac{\Gamma L\|s_k\|}{\gamma^2} \leq \frac{\Gamma L D}{\gamma^2},$$

where $D = \max\{\|y - z\| : y, z \in S\}$ is the diameter of the level set S. On the other hand,

$$|\frac{g_{k+1}^T g_{k+1}}{y_k^T s_k}| \leq \frac{\Gamma^2}{(1-\sigma)\omega\alpha_k\gamma^2}.$$

Therefore,

$$|\beta_k^N| \leq \frac{\Gamma L D}{\gamma^2} + \frac{\Gamma^2}{(1-\sigma)\omega\alpha_k\gamma^2} \equiv E. \tag{25}$$

Now, we can write

$$\|d_{k+1}\| \leq \|g_{k+1}\| + |\beta_k^N|\|s_k\| \leq \Gamma + E D. \tag{26}$$

Since the level set L is bounded and the function f is bounded from below, using Lemma 4.2, from (4) it follows that

$$0 < \sum_{k=0}^{\infty} \frac{(g_k^T d_k)^2}{\|d_k\|^2} < \infty, \tag{27}$$

i.e. the Zoutendijk condition holds. Therefore, from Theorem 2.2 using (27) the descent property yields:

$$\sum_{k=0}^{\infty} \frac{\gamma^4}{\|d_k\|^2} \leq \sum_{k=0}^{\infty} \frac{\|g_k\|^4}{\|d_k\|^2} \leq \sum_{k=0}^{\infty} \frac{1}{\omega^2} \frac{(g_k^T d_k)^2}{\|d_k\|^2} < \infty,$$

which contradicts (26). Hence, $\gamma = \liminf_{k\to\infty} \|g_k\| = 0$. □

Therefore, when $0 < \theta_k < 1$, our conjugate gradient algorithm is globally convergent, meaning that either $g_k = 0$ for some k or (22) holds. Observe that, in the conditions of Theorem 2.2, the direction d_{k+1} satisfies the sufficient descent condition independently of the line search.

5 Numerical Experiments

In this section, we present the computational performance of a Fortran implementation of the CCOMB algorithm on a set of 750 unconstrained optimization test problems. These are the unconstrained problems in the CUTE library [14], along with

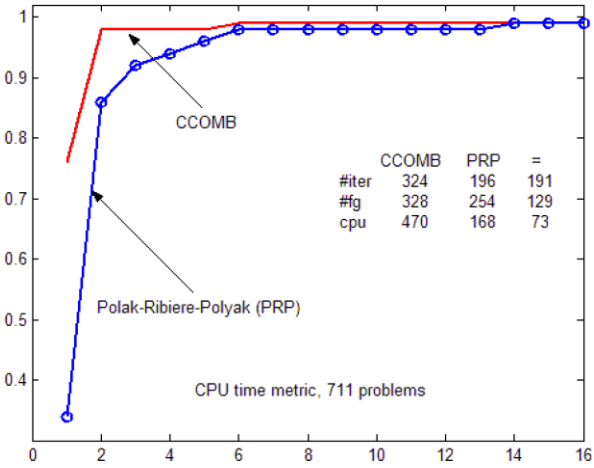

Fig. 1 Performance based on CPU time. CCOMB versus Polak-Ribière-Polyak (PRP)

other large-scale optimization problems presented in [15]. We selected 75 large-scale unconstrained optimization problems in extended or generalized form. Each of them is tested 10 times for a gradually increasing number of variables $n = 1000, 2000, \ldots, 10\,000$. At the same time, we present comparisons with other conjugate gradient algorithms, including the performance profiles of Dolan and Moré [13].

All algorithms implement the Wolfe line search conditions with $\rho = 0.0001$ and $\sigma = 0.9$ and the same stopping criterion $\|g_k\|_\infty \leq 10^{-6}$, where $\|.\|_\infty$ is the maximum absolute component of a vector.

The comparisons of algorithms are given in the following context. Let f_i^{ALG1} and f_i^{ALG2} be the optimal value found by ALG1 and ALG2 for problem $i = 1, \ldots, 750$, respectively. We say that in the particular problem i the performance of ALG1 was better than the performance of ALG2 if

$$|f_i^{ALG1} - f_i^{ALG2}| < 10^{-3} \tag{28}$$

and the number of iterations or the number of function-gradient evaluations is smaller or the CPU time of ALG1 is less than the number of iterations or the number of function-gradient evaluations or the CPU time corresponding to ALG2, respectively.

All codes are written in double precision Fortran and compiled with f77 (default compiler settings) on an Intel Pentium 4, 1.8 GHz workstation. All these codes are authored by Andrei.

In the first set of numerical experiments, we compare the performance of CCOMB to the PRP and the DY conjugate gradient algorithms. Figures 1 and 2 show the Dolan and Moré CPU performance profiles of CCOMB versus PRP and of CCOMB versus DY, respectively.

When comparing CCOMB to PRP (Fig. 1) subject to the number of iterations, we see that CCOMB was better in 324 problems (i.e. it achieved the minimum number of iterations in 324 problems), PRP was better in 196 problems and they achieved the same number of iterations in 191 problems, etc. Out of 750 problems, only for 711 problems does the criterion (28) hold. Similarly, in Fig. 2 we see the number of

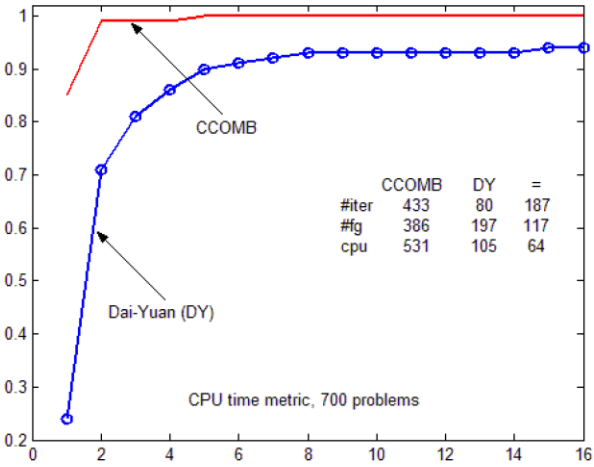

Fig. 2 Performance based on CPU time. CCOMB versus Dai-Yuan (DY)

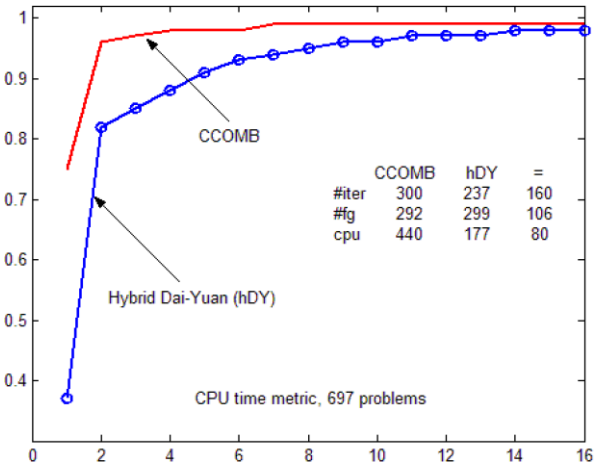

Fig. 3 Performance based on CPU time. CCOMB versus hybrid Dai-Yuan (hDY)

problems for which CCOMB is better than DY. Observe that the convex combination of PRP and DY expressed as in (7) is far more successful than PRP or DY algorithms.

The second set of numerical experiments refers to the comparisons of CCOMB to the hybrid conjugate gradient algorithms hDY, hDYz, GN, HuS, TS and LS-CD. Figures 3, 4, 5, 6, 7, and 8 present the Dolan and Moré CPU performance profiles of these algorithms as well as the number of problems solved by each algorithm in minimum number of iterations, minimum number of function evaluations and minimum CPU time, respectively.

From the figures, we can see that CCOMB is the top performer. Since these codes use the same Wolfe line search and the same stopping criterion, they differ in their choice of the search direction. Hence, among these conjugate gradient algorithms, CCOMB appears to generate the best search direction.

In the third set of numerical experiments, we compare CCOMB to the CG_DESCENT conjugate gradient algorithm by Hager and Zhang [30] with the Wolfe line search.

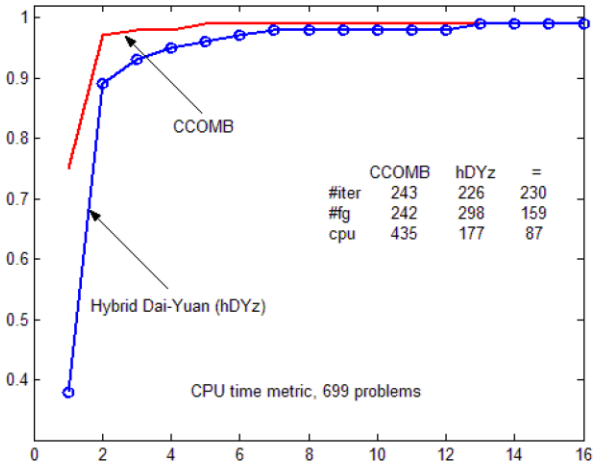

Fig. 4 Performance based on CPU time. CCOMB versus hybrid Dai-Yuan (hDYz)

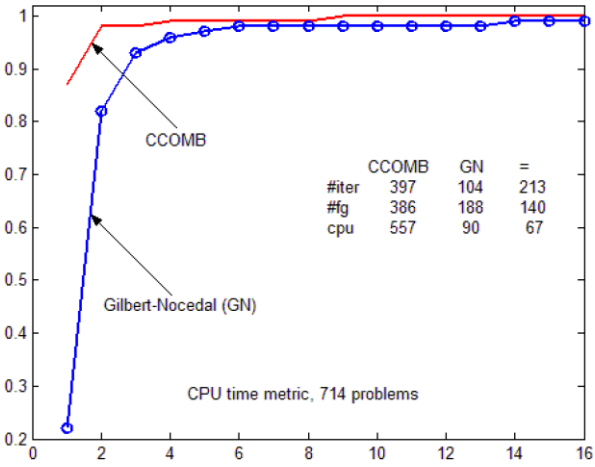

Fig. 5 Performance based on CPU time. CCOMB versus Gilbert-Nocedal (GN)

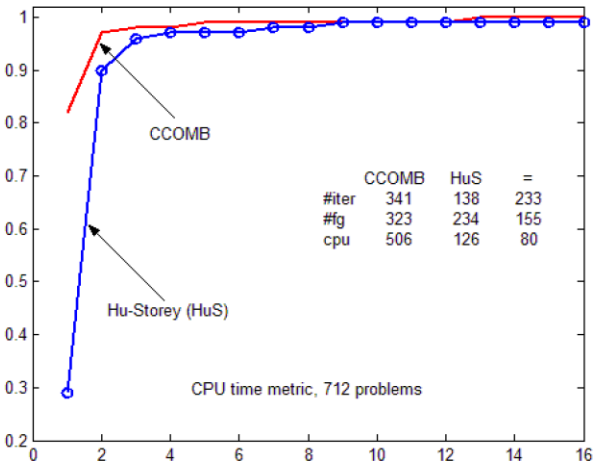

Fig. 6 Performance based on CPU time. CCOMB versus Hu-Storey (HuS)

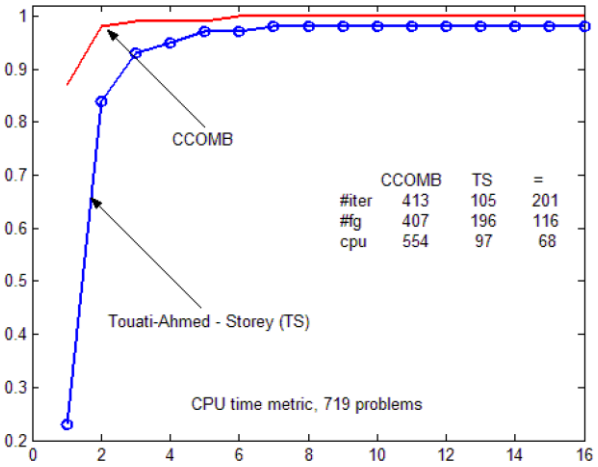

Fig. 7 Performance based on CPU time. CCOMB versus Touati-Ahmed-Storey (TS)

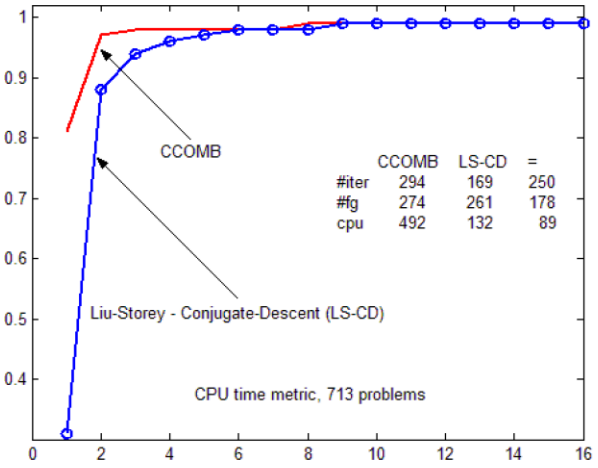

Fig. 8 Performance based on CPU time. CCOMB versus Liu-Storey-Conjugate Descent (LS-CD)

The computational scheme implemented in CG_DESCENT is a modification of the Hestenes and Stiefel method which satisfies the sufficient descent condition, independently of the accuracy of the line search. Figure 9 presents the performance profile of CCOMB versus CG_DESCENT. We see that CG_DESCENT is more robust than CCOMB.

6 Conclusions

There exists a large variety of conjugate gradient algorithms. In this paper, we have presented a new hybrid conjugate gradient algorithm in which the parameter β_k is computed as a convex combination of β_k^{PRP} and β_k^{DY}, i.e. $\beta_k^N = (1-\theta_k)\beta_k^{PRP} + \theta_k\beta_k^{DY}$. The parameter θ_k is obtained from the conjugacy condition. For uniformly convex functions, if the stepsize s_k approaches zero, the gradient is bounded in the sense that $\eta_1\|s_k\|^2 \leq \|g_k\|^2 \leq \eta_2\|s_{k-1}\|$ and the line search satisfies the strong Wolfe

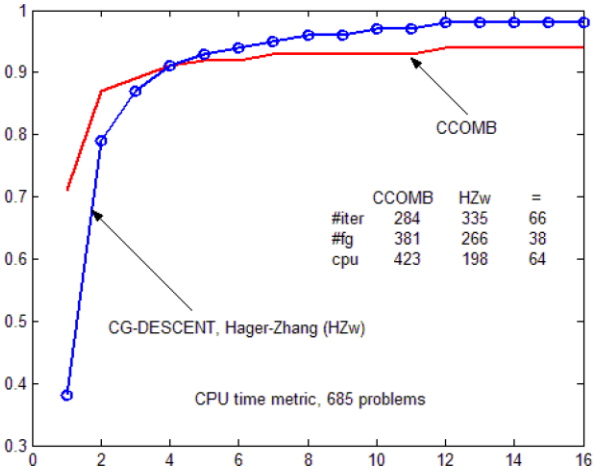

Fig. 9 Performance based on CPU time. CCOMB versus CG_DESCENT with Wolfe line search (HZw)

conditions, then our hybrid conjugate gradient algorithm is globally convergent. For general nonlinear functions, if the parameter θ_k from β_k^N definition is bounded, then our hybrid conjugate gradient is globally convergent. The performance profile of our algorithm is higher than those of the well established conjugate gradient algorithms for a test set consisting of 750 unconstrained optimization problems, some of them from the CUTE library. Additionally, the proposed hybrid conjugate gradient algorithm is more robust than the PRP and the DY conjugate gradient algorithms. However, CG_DESCENT is more robust than the CCOMB algorithm.

References

1. Hager, W.W., Zhang, H.: A survey of nonlinear conjugate gradient methods. Pac. J. Optim. **2**, 35–58 (2006)
2. Fletcher, R., Reeves, C.: Function minimization by conjugate gradients. Comput. J. **7**, 149–154 (1964)
3. Dai, Y.H., Yuan, Y.: A nonlinear conjugate gradient method with a strong global convergence property. SIAM J. Optim. **10**, 177–182 (1999)
4. Fletcher, R.: Unconstrained Optimization. Practical Methods of Optimization, vol. 1. Wiley, New York (1987)
5. Polak, E., Ribière, G.: Note sur la convergence de directions conjuguée. Rev. Fr. Inf. Rech. Oper. 3e Année **16**, 35–43 (1969)
6. Poliak, B.T.: The conjugate gradient method in extreme problems. USSR Comput. Math. Math. Phys. **9**, 94–112 (1969)
7. Hestenes, M.R., Stiefel, E.L.: Methods of conjugate gradients for solving linear systems. J. Res. Natl. Bur. Stand. **49**, 409–436 (1952)
8. Liu, Y., Storey, C.: Efficient generalized conjugate gradient algorithms, Part 1: Theory. J. Optim. Theory Appl. **69**, 129–137 (1991)
9. Touati-Ahmed, D., Storey, C.: Efficient hybrid conjugate gradient techniques. J. Optim. Theory Appl. **64**, 379–397 (1990)
10. Hu, Y.F., Storey, C.: Global convergence result for conjugate gradient methods. J. Optim. Theory Appl. **71**, 399–405 (1991)
11. Gilbert, J.C., Nocedal, J.: Global convergence properties of conjugate gradient methods for optimization. SIAM J. Optim. **2**, 21–42 (1992)
12. Dai, Y.H., Yuan, Y.: An efficient hybrid conjugate gradient method for unconstrained optimization. Ann. Oper. Res. **103**, 33–47 (2001)

13. Dolan, E.D., Moré, J.J.: Benchmarking optimization software with performance profiles. Math. Program. **91**, 201–213 (2002)
14. Bongartz, I., Conn, A.R., Gould, N.I.M., Toint, P.L.: CUTE: constrained and unconstrained testing environments. ACM Trans. Math. Softw. **21**, 123–160 (1995)
15. Andrei, N.: An unconstrained optimization test functions collection. Adv. Model. Optim. **10**, 147–161 (2008)
16. Powell, M.J.D.: Restart procedures of the conjugate gradient method. Math. Program. **2**, 241–254 (1977)
17. Yuan, Y.: Analysis on the conjugate gradient method. Optim. Methods Softw. **2**, 19–29 (1993)
18. Powell, M.J.D.: Nonconvex minimization calculations and the conjugate gradient method. In: Numerical Analysis, Dundee, 1983. Lecture Notes in Mathematics, vol. 1066, pp. 122–141. Springer, Berlin (1984)
19. Dai, Y.H.: Analysis of conjugate gradient methods. Ph.D. Thesis, Institute of Computational Mathematics and Scientific/Engineering Computing, Chinese Academy of Science (1997)
20. Dai, Y.H.: New properties of a nonlinear conjugate gradient method. Numer. Math. **89**, 83–98 (2001)
21. Dai, Y.H., Liao, L.Z., Li, D.: On restart procedures for the conjugate gradient method. Numer. Algorithms **35**, 249–260 (2004)
22. Shanno, D.F., Phua, K.H.: Algorithm 500, Minimization of unconstrained multivariate functions. ACM Trans. Math. Softw. **2**, 87–94 (1976)
23. Birgin, E., Martínez, J.M.: A spectral conjugate gradient method for unconstrained optimization. Appl. Math. Optim. **43**, 117–128 (2001)
24. Andrei, N.: Scaled conjugate gradient algorithms for unconstrained optimization. Comput. Optim. Appl. **38**, 401–416 (2007)
25. Andrei, N.: Scaled memoryless BFGS preconditioned conjugate gradient algorithm for unconstrained optimization. Optim. Methods Softw. **22**, 561–571 (2007)
26. Andrei, N.: A scaled BFGS preconditioned conjugate gradient algorithm for unconstrained optimization. Appl. Math. Lett. **20**, 645–650 (2007)
27. Kiwiel, K.C., Murty, K.: Convergence of the steepest descent method for minimizing quasiconvex functions. J. Optim. Theory Appl. **89**(1), 221–226 (1996)
28. Dai, Y.H., Han, J.Y., Liu, G.H., Sun, D.F., Yin, X., Yuan, Y.: Convergence properties of nonlinear conjugate gradient methods. SIAM J. Optim. **10**, 348–358 (1999)
29. Dai, Y.H., Liao, L.Z.: New conjugacy conditions and related nonlinear conjugate gradient methods. Appl. Math. Optim. **43**, 87–101 (2001)
30. Hager, W.W., Zhang, H.: A new conjugate gradient method with guaranteed descent and an efficient line search. SIAM J. Optim. **16**, 170–192 (2005)