New hybrid conjugate gradient algorithms for unconstrained optimization

Neculai Andrei
Research Institute for Informatics,
Center for Advanced Modeling and Optimization,
8-10, Averescu Avenue, Bucharest 1, Romania,
E-mail: nandrei@ici.ro

Abstract. New hybrid conjugate gradient algorithms are proposed and analyzed. In these hybrid algorithms the famous parameter β_k is computed as a convex combination of the Polak-Ribière-Polyak and Dai-Yuan conjugate gradient algorithms. In one hybrid algorithm the parameter in convex combination is computed in such a way that the conjugacy condition is satisfied, independent of the line search. In the other, the parameter in convex combination is computed in such a way that the conjugate gradient direction is the Newton direction. The algorithm uses the standard Wolfe line search conditions. Numerical comparisons with conjugate gradient algorithms using a set of 750 unconstrained optimization problems, some of them from the CUTE library, show that the hybrid computational scheme based on conjugacy condition outperform the known hybrid conjugate gradient algorithms.

MSC: 49M07, 49M10, 90C06, 65K
Keywords: Unconstrained optimization, hybrid conjugate gradient method, conjugacy condition, numerical comparisons

1. Introduction

Let us consider the nonlinear unconstrained optimization problem

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

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 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 $\|\cdot\|$ the Euclidean norm and define $y_k = g_{k+1} - g_k$. The line search in the conjugate gradient algorithms often is based on the standard Wolfe conditions:

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

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

where d_k is a descent direction and $0 < \rho \le \sigma < 1$. Plenty of conjugate gradient methods are known, and an excellent survey of these methods, with a special attention on their global convergence, is given by Hager and Zhang [19]. Different conjugate gradient algorithms correspond to different choices for the scalar parameter β_k. Some of these methods as

Fletcher and Reeves (FR) [16], Dai and Yuan (DY) [12] and Conjugate Descent (CD) proposed by Fletcher [15]:

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

have strong convergence properties, but they may have modest practical performance due to jamming. On the other hand, the methods of Polak – Ribière [23] and Polyak (PRP) [24], Hestenes and Stiefel (HS) [20] or Liu and Storey (LS) [22]:

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

in general may not be convergent, but they often have better computational performances.

In this paper we focus on hybrid conjugate gradient methods. These methods are combinations of different conjugate gradient algorithms, mainly they being proposed to avoid the jamming phenomenon. One of the first hybrid conjugate gradient algorithms has been introduced by Touati-Ahmed and Storey [28], where the parameter β_k is computed as:

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

The PRP method has a built-in restart feature that directly addresses to jamming. 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 [21], where β_k in (3) is:

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

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

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

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

The CD method of Fletcher [15] is very close to FR method. With an exact line search, CD method is identical to FR. Similarly, for an exact line search, LS method is also identical to PRP. Therefore, the hybrid LS-CD method with an exact line search has similar performances with the hybrid method of Hu and Storey.

Gilbert and Nocedal [17] suggested a combination between PRP and FR methods as:

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

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 continuously the method is global convergent. In an effort to improve their algorithm, Dai and Yuan [13] combined their algorithm with other conjugate gradient algorithm, proposing the following two hybrid methods:

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

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

where $c = (1-\sigma)/(1+\sigma)$. For the standard Wolfe conditions (4) and (5), under the Lipschitz continuity of the gradient, Dai and Yuan [13] established the global convergence of these hybrid computational schemes.

In this paper we propose another hybrid conjugate gradient as a convex combination of PRP and DY conjugate gradient algorithms. We selected these two methods to combine in a hybrid conjugate gradient algorithm because PRP has good computational properties, on one side, and DY has strong convergence properties, on the other side. Often PRP method performs better in practice than DY and we speculate this in order to have a good practical conjugate algorithm. The structure of the paper is as follows. In section 2 we introduce our hybrid conjugate gradient algorithm and prove that it generates descent directions satisfying in some conditions the sufficient descent condition. Section 3 presents the algorithms and in section 4 we show its convergence analysis. In section 5 some numerical experiments and performance profiles of Dolan-Moré [14] corresponding to this new hybrid conjugate gradient algorithm and 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 [6], as well as some other unconstrained optimization problems presented in [1] show that this hybrid conjugate gradient algorithm outperform the known hybrid conjugate gradient algorithms.

2. New hybrid conjugate gradient algorithms

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), and 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 follows to be determined. 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 [23] proved that when 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 [25] showed that if the step length $s_k = x_{k+1} - x_k$ approaches to 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 direction, Yuan [29] 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 [26] gave a 3 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 [7] 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 [17] proved that the PRP method is one of the best conjugate gradient methods, and this is the main motivation to consider it in (7).

On the other hand, the DY method always generates descent directions, and in [8] Dai 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 $\gamma_1 \le \|g_k\| \le \gamma_2$ for all k, then for any $p \in (0,1)$, there exists a constant $c > 0$ such that the sufficient descent condition $g_i^T d_i \le -c\|g_i\|^2$ holds for at least $\lfloor pk \rfloor$ indices $i \in [0,k]$, where $\lfloor j \rfloor$ denotes the largest integer $\le j$. Therefore, this property is the main reason we consider DY method in (7)

It easy to see that:

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

Supposing that d_k is a descent direction ($d_0 = -g_0$), then for the algorithm given by (2) and (8) we can prove the following result.

Theorem 1. *Assume that α_k in algorithm (2) and (8) is determined by Wolfe line search (4) and (5). If $0 < \theta_k < 1$, and*

$$\left|\frac{g_k^T s_k}{y_k^T s_k}\right| \|g_{k+1}\|^2 \ge \frac{(g_{k+1}^T y_k)(g_{k+1}^T s_k)}{\|g_k\|^2}, \tag{9}$$

then direction d_{k+1} given by (8) is a descent direction.

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

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

But, $y_k^T s_k > 0$ by (5) and since $g_k^T s_k \le 0$, it follows that

$$\frac{g_k^T s_k}{y_k^T s_k}\|g_{k+1}\|^2 \le 0 .$$

Therefore, from (9), it follows that $g_{k+1}^T d_{k+1} \le 0$, i.e. the direction d_{k+1} is a descent one. ■

Theorem 2. *Suppose that $(g_{k+1}^T y_k)(g_{k+1}^T s_k) \le 0$. If $0 < \theta_k < 1$ then the direction d_{k+1} given by (8) satisfies the sufficient descent condition*

$$g_{k+1}^T d_{k+1} \le -\left(1 - \theta_k \frac{g_{k+1}^T s_k}{y_k^T s_k}\right)\|g_{k+1}\|^2 . \tag{10}$$

Proof. From (8) we have:

$$g_{k+1}^T d_{k+1} = -\|g_{k+1}\|^2 + (1-\theta_k)\frac{g_{k+1}^T y_k}{g_k^T g_k} g_{k+1}^T s_k + \theta_k \frac{g_{k+1}^T g_{k+1}}{y_k^T s_k} g_{k+1}^T s_k$$

$$= -\|g_{k+1}\|^2 + \theta_k \frac{g_{k+1}^T s_k}{y_k^T s_k}\|g_{k+1}\|^2 + (1-\theta_k)\frac{(g_{k+1}^T y_k)(g_{k+1}^T s_k)}{g_k^T g_k}$$

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

Observe that, 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 $y_k^T s_k / g_{k+1}^T s_k > 1$. Therefore, if $0 < \theta_k < 1$, it follows that $\theta_k < y_k^T s_k / g_{k+1}^T s_k$. Therefore

$$1-\theta_k \frac{g_{k+1}^T s_k}{y_k^T s_k} > 0$$

proving the theorem. ■

To select the parameter θ_k we consider the following two possibilities. In the first hybrid conjugate gradient algorithm the parameter θ_k is selected in such a manner that the conjugacy condition $y_k^T d_{k+1} = 0$ is satisfied at every iteration, independent on the line search. Hence, from $y_k^T d_{k+1} = 0$ after some algebra, using (8), we get:

$$\theta_k = \frac{(y_k^T g_{k+1})(y_k^T s_k) - (y_k^T g_{k+1})(g_k^T g_k)}{(y_k^T g_{k+1})(y_k^T s_k) - \|g_{k+1}\|^2 \|g_k\|^2}. \tag{11}$$

In the second algorithm the parameter θ_k is selected in such a manner that the direction d_{k+1} from (8) is the Newton direction, i.e.

$$-\nabla^2 f(x_{k+1})^{-1} g_{k+1} = -g_{k+1} + (1-\theta_k)\frac{y_k^T g_{k+1}}{g_k^T g_k} s_k + \theta_k \frac{g_{k+1}^T g_{k+1}}{y_k^T s_k} s_k. \tag{12}$$

Having in view that $\nabla^2 f(x_{k+1}) s_k = y_k$, from (12) we get:

$$\theta_k = \frac{(y_k^T g_{k+1} - s_k^T g_{k+1})\|g_k\|^2 - (g_{k+1}^T y_k)(y_k^T s_k)}{\|g_{k+1}\|^2 \|g_k\|^2 - (g_{k+1}^T y_k)(y_k^T s_k)}. \tag{13}$$

Observe that the parameter θ_k given by (11) or (13) can be outside the interval $[0,1]$. However, in order to have a real convex combination in (7) the following rule is considered: if $\theta_k \le 0$, then set $\theta_k = 0$ in (7), i.e. $\beta_k^N = \beta_k^{PRP}$; if $\theta_k \ge 1$, 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 PRP and DY algorithms.

3. The New Hybrid Algorithms (CCOMB, NDOMB)

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

Step 2. Test for continuation of iterations. If $\|g_k\|_\infty \le 10^{-6}$, 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 $(y_k^T g_{k+1})(y_k^T s_k) - \|g_{k+1}\|^2 \|g_k\|^2 = 0$, then set $\theta_k = 0$, otherwise compute θ_k as follows:

CCOMB algorithm (θ_k from Conjugacy Condition):

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

NDOMB algorithm (θ_k from Newton Direction):

$$\theta_k = \frac{(y_k^T g_{k+1} - s_k^T g_{k+1})\|g_k\|^2 - (g_{k+1}^T y_k)(y_k^T s_k)}{\|g_{k+1}\|^2 \|g_k\|^2 - (g_{k+1}^T y_k)(y_k^T s_k)}.$$

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

$$\left|g_{k+1}^T g_k\right| \geq 0.2\|g_{k+1}\|^2, \tag{14}$$

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, 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 [10], 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 or is equal to the Newton direction. Under reasonable assumptions, conditions (4), (5) and (14) are sufficient to prove the global convergence of the algorithm. We consider this aspect in the next section.

The first trial of the step length crucially affects the practical behavior of the algorithm. At every iteration $k \geq 1$ the starting guess for the step α_k in the line search is computed as $\alpha_{k-1}\|d_{k-1}\|_2 / \|d_k\|_2$. This selection was considered for the first time by Shanno and Phua in CONMIN [27]. It is also considered in the packages: SCG by Birgin and Martínez [5] and in SCALCG by Andrei [2,3,4].

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 $x \in S$.

In [11] it is proved that for any conjugate gradient method with strong Wolfe line search the following general holds:

Lemma 1. *Suppose that the 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*

$$\sum_{k\geq 1}\frac{1}{\|d_k\|^2}=\infty, \tag{15}$$

then

$$\liminf_{k\to\infty}\|g_k\|=0. \ \blacksquare \tag{16}$$

For uniformly convex functions which satisfy the above assumptions we can prove that the norm of d_{k+1} generated by (8) is bounded above. Thus, by lemma 1 we have the following result.

Theorem 3. *Suppose that the assumptions (i) and (ii) hold. Consider the algorithm (2) and (8), where* d_{k+1} *is a descent direction and* α_k *is obtained by the strong Wolfe line search.*

$$f(x_k+\alpha_k d_k)-f(x_k)\leq \rho\alpha_k g_k^T d_k, \tag{17}$$

$$\left|g_{k+1}^T d_k\right|\leq -\sigma g_k^T d_k. \tag{18}$$

If for $k\geq 0$, $\|s_k\|$ *tends to zero and there exists the nonnegative constants* η_1 *and* η_2 *such that*

$$\|g_k\|^2\geq \eta_1\|s_k\|^2 \quad \textit{and} \quad \|g_{k+1}\|^2\leq \eta_2\|s_k\|, \tag{19}$$

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

$$(\nabla f(x)-\nabla f(y))^T(x-y)\geq \mu\|x-y\|^2, \tag{20}$$

then

$$\lim_{k\to\infty} g_k=0. \tag{21}$$

Proof. From (20) 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 (19) we have:

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

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

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

Hence,

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

which implies that (15) is true. Therefore, by lemma 1 we have (16), which for uniformly convex functions is equivalent to (21). ■

Powell [25] showed that for general functions the PRP method is globally convergent if the steplengths $\|s_k\|=\|x_{k+1}-x_k\|$ tend to zero, i.e. $\|s_k\|\leq\|s_{k-1}\|$ is a condition of convergence. For convergence of our algorithms from (19) we see that along the iterations, 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 gradient can satisfy (19).

In the numerical experiments we observed that (19) is always 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 [17], Dai and Liao [9] and that of Hager and Zhang [18]. Global convergence proof of these new hybrid conjugate gradient algorithms 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 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*

$$\alpha_k \ge \frac{1-\sigma}{L}\frac{|g_k^T d_k|}{\|d_k\|^2}. \tag{22}$$

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

$$(\sigma-1)g_k^T d_k \le (g_{k+1}-g_k)^T d_k \le L\alpha_k \|d_k\|^2. \tag{23}$$

Since d_k is a descent direction and $\sigma<1$, (22) follows immediately from (23). ■

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

$$\liminf_{k\to\infty}\|g_k\|=0. \tag{24}$$

Proof. By the Wolfe condition (5) we have:

$$y_k^T s_k=(g_{k+1}-g_k)^T s_k \ge (\sigma-1)g_k^T s_k=-(1-\sigma)g_k^T s_k. \tag{25}$$

By Theorem 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,

$$-g_k^T d_k \ge \omega\|g_k\|^2. \tag{26}$$

Combining (25) with (26) 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

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

But,

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

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

On the other hand,

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

Therefore,

$$\left|\beta_k^N\right| \le \frac{\Gamma LD}{\gamma^2} + \frac{\Gamma^2}{(1-\sigma)\omega\alpha_k\gamma^2} \equiv E\,. \tag{27}$$

Now, we can write

$$\|d_{k+1}\| \le \|g_{k+1}\| + \left|\beta_k^N\right|\|s_k\| \le \Gamma + ED. \tag{28}$$

Since the level set L is bounded and the function f is bounded from below, using Lemma 2, from (4) it follows that

$$0 < \sum_{k=0}^{\infty} \frac{(g_k^T d_k)^2}{\|d_k\|^2} < \infty\,, \tag{29}$$

i.e. the Zoutendijk condition holds. Therefore, from Theorem 2 using (29), the descent property yields:

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

which contradicts (28). Hence, $\gamma = \liminf_{k\to\infty} \|g_k\| = 0$. ■

Therefore, when $0 < \theta_k \le 1$ our hybrid conjugate gradient algorithms are globally convergent, meaning that either $g_k = 0$ for some k or (24) holds. Observe that in conditions of Theorem 2 the direction d_{k+1} satisfies the sufficient descent condition independent on the line search.

5. Numerical experiments

In this section we present the computational performance of a Fortran implementation of the CCOMB and NDOMB algorithms on a set of 750 unconstrained optimization test problems. The test problems are the unconstrained problems in the CUTE [6] library, along with other large-scale optimization problems presented in [1]. We selected 75 large-scale unconstrained optimization problems in extended or generalized form. Each problem is tested 10 times for a gradually increasing number of variables: $n = 1000, 2000, \ldots, 10000$. At the same time we present comparisons with other conjugate gradient algorithms, including the performance profiles of Dolan and Moré [14].

All algorithms implement the Wolfe line search conditions with $\rho = 0.0001$ and $\sigma = 0.9$, and the same stopping criterion $\|g_k\|_\infty \le 10^{-6}$, 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:

$$\left| f_i^{ALG1} - f_i^{ALG2} \right| < 10^{-3} \tag{30}$$

and the number of iterations, or the number of function-gradient evaluations, or the CPU time of ALG1 was 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, 1.8GHz workstation. All these codes are authored by Andrei. The performances of these algorithms have been evaluated using the profiles of Dolan and Moré [14]. That is, for each algorithm we plot the fraction of problems for which the algorithm is within a factor of the best CPU time. The left side of these Figures gives the percentage of the test problems, out of 750, for which an algorithm is more performant; the right side gives the percentage of the test problems that were successfully solved by each of the algorithms. Mainly, the right side represents a measure of an algorithm's robustness.

In the first set of numerical experiments we compare the performance of CCOMB to NDOMB. Figure 1 shows the Dolan and Moré CPU performance profiles of CCOMB versus NDOMB.

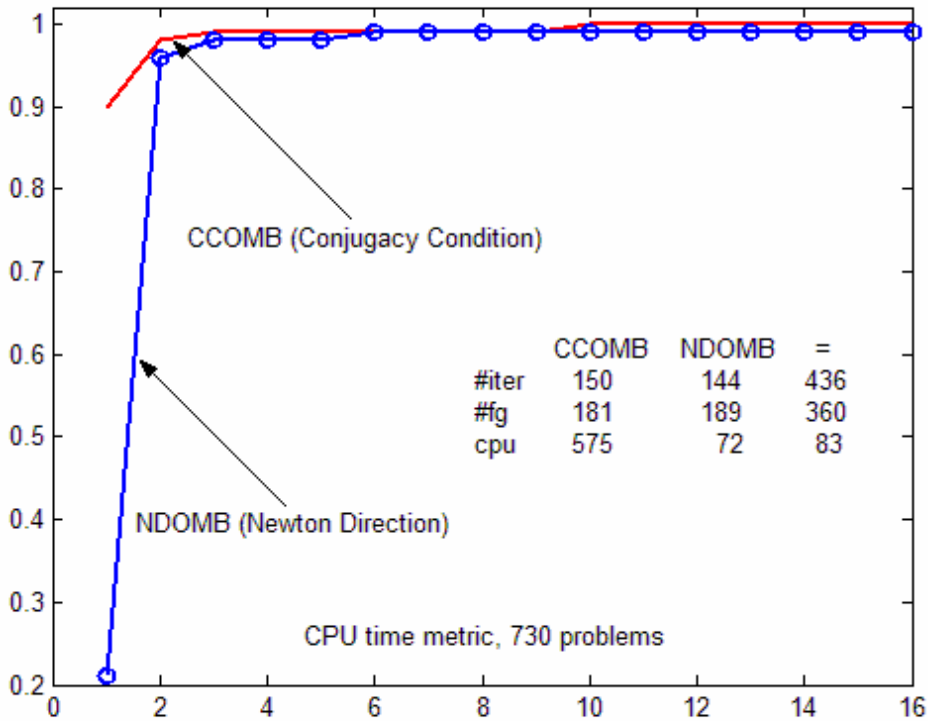

Fig. 1. Performance based on CPU time. CCOMB versus NDOMB.

Observe that CCOMB outperforms NDOMB in the vast majority of problems. Only 730 problems out 750 satisfy the criterion (30). Referring to the CPU time, CCOMB was better in 575 problems, in contrast with NDOMB which solved only 72 problems in a better CPU time.

In the second set of numerical experiments we compare the performance of CCOMB to the PRP and DY conjugate gradient algorithms. Figures 2 and 3 show the Dolan and Moré CPU performance profiles of CCOMB versus PRP and CCOMB versus DY, respectively.

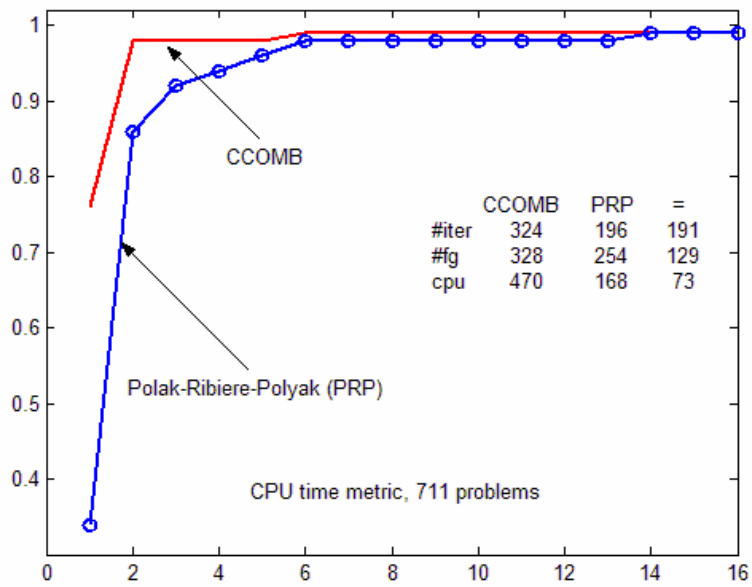

Fig. 2. Performance based on CPU time. CCOMB versus Polak-Ribière-Polyak (PRP).

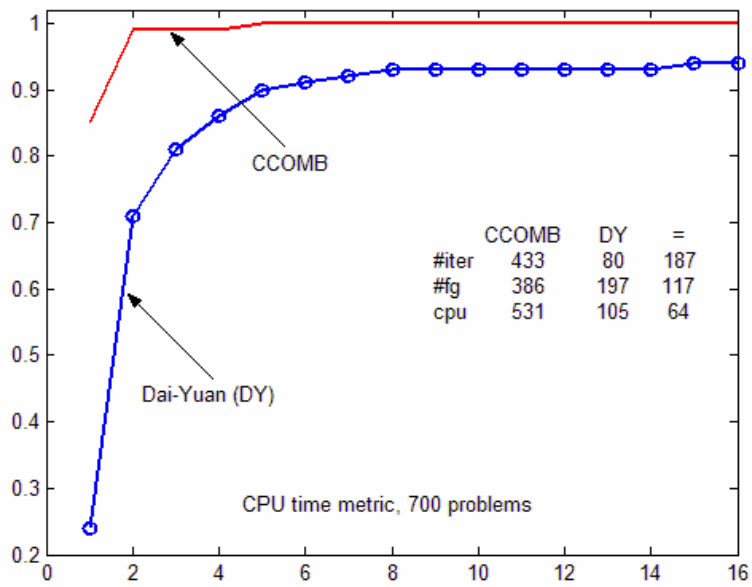

Fig. 3. Performance based on CPU time. CCOMB versus Dai-Yuan (DY).

When comparing CCOMB to PRP (Figure 2), 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 the criteria (30) holds. Similarly, in Figure 3 we see the number of problems for which CCOMB was 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 third set of numerical experiments refers to the comparisons of CCOMB to hybrid conjugate gradient algorithms: hDY, hDYz, GN, HuS, TS and LS-CD. Figures 4-9 presents the Dolan and Moré CPU performance profiles of these algorithms, as well as the

number of problems solved by each algorithms in minimum number of iterations, minimum number of function evaluations and minimum CPU time, respectively.

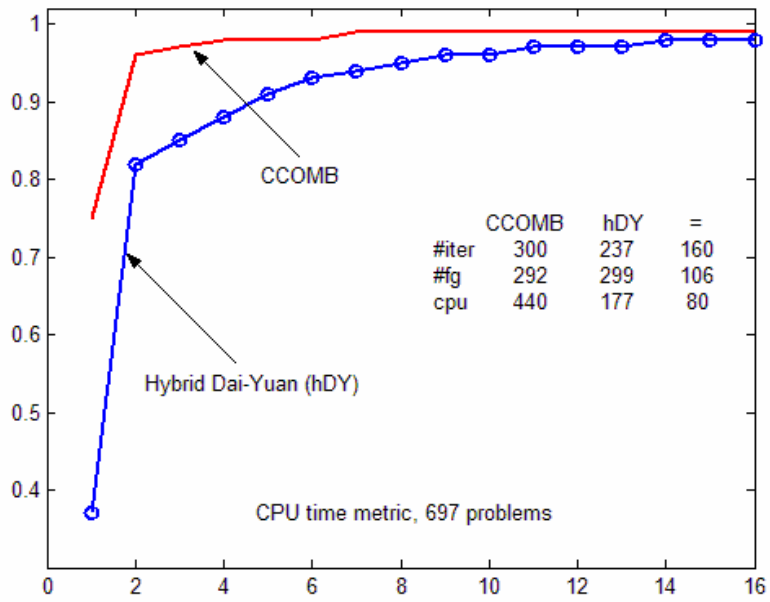

Fig. 4. Performance based on CPU time. CCOMB versus hybrid Dai-Yuan (hDY).

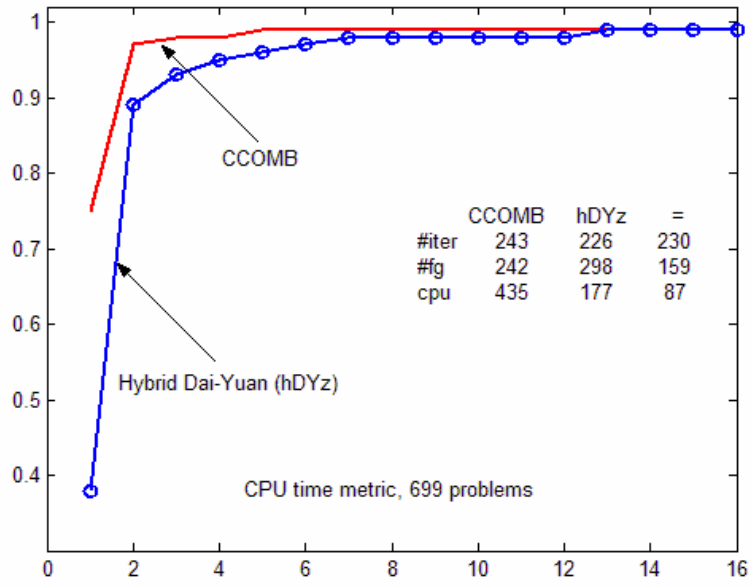

Fig. 5. Performance based on CPU time. CCOMB versus hybrid Dai-Yuan (hDYz).

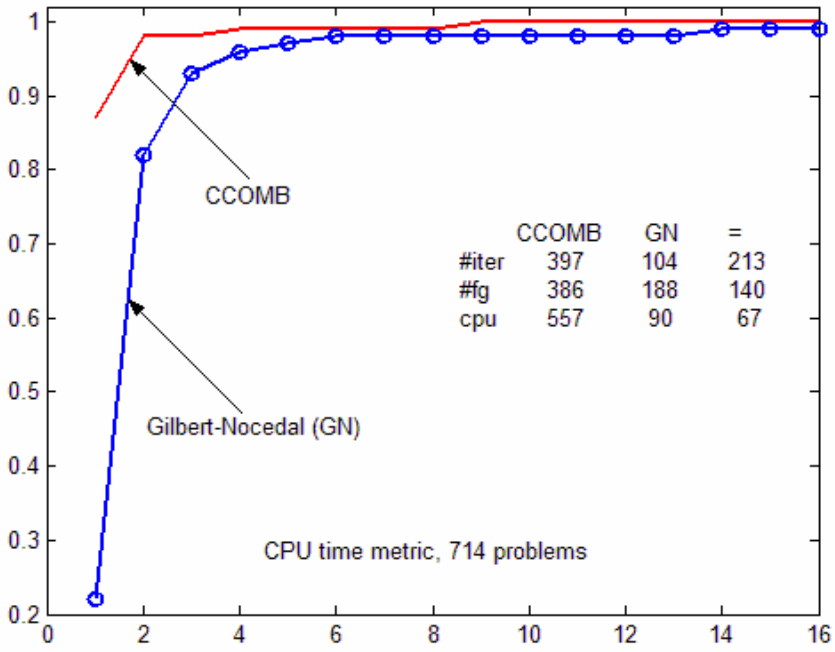

Fig. 6. Performance based on CPU time. CCOMB versus Gilbert-Nocedal (GN).

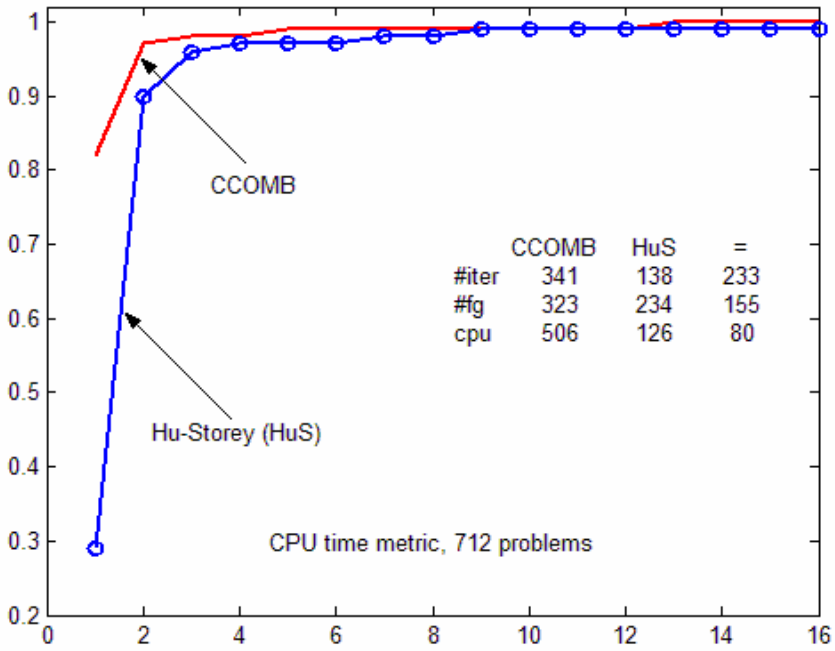

Fig. 7. Performance based on CPU time. CCOMB versus Hu-Storey (HuS).

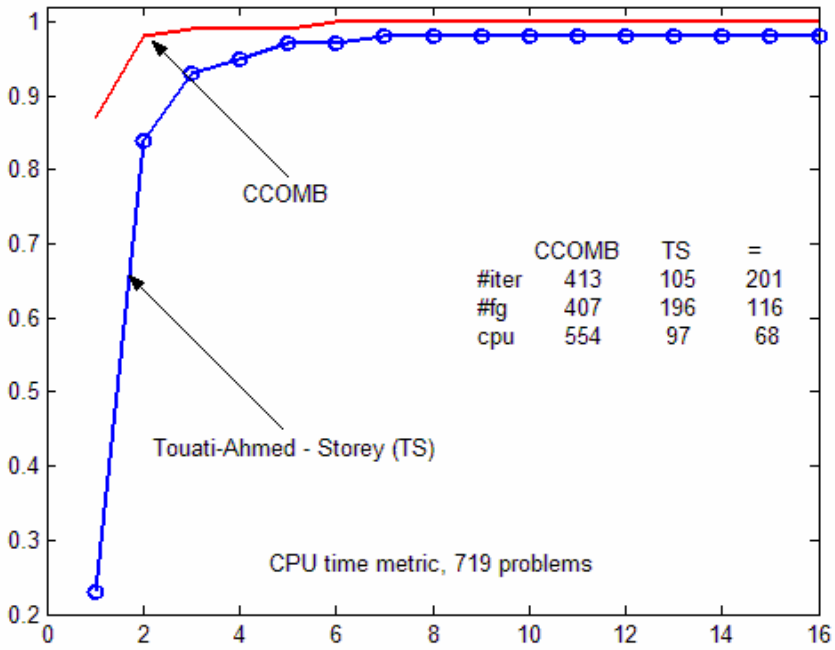

Fig. 8. Performance based on CPU time. CCOMB versus Touati-Ahmed - Storey (TS).

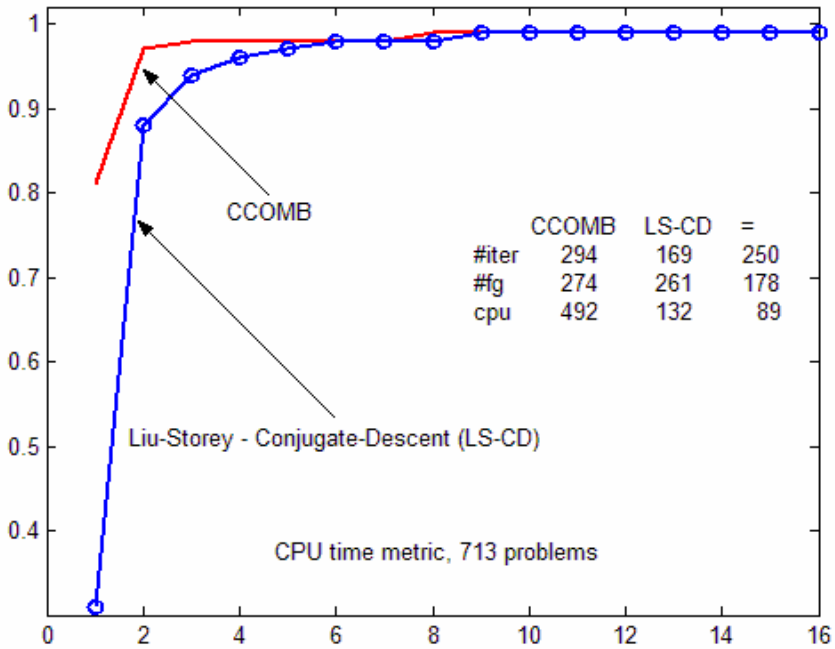

Fig. 9. Performance based on CPU time. CCOMB versus Liu-Storey – Conjugate Descent (LS-CD).

From these Figures above we see that CCOMB is 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 we considered here, CCOMB appears to generate the best search direction.

In the fourth set of numerical experiments we compare CCOMB to CG_DESCENT conjugate gradient algorithm of Hager and Zhang [18]. The computational scheme implemented in CG_DESCENT is a modification of the Hestenes and Stiefel method which

satisfies the sufficient descent condition, independent of the accuracy of the line search. The CG_DESCENT code, authored by Hager and Zhang, contains the variant CG_DESCENT (HZw) implementing the Wolfe line search and the variant CG_DESCENT (HZaw) implementing an approximate Wolfe line search. There are two main points associated to CG_DESCENT. Firstly, the scalar products are implemented using the loop unrolling of depth 5. This is efficient for large-scale problems (over 10^6 variables). Secondly, the Wolfe line search is implemented using a very fine numerical interpretation of the first Wolfe condition (4). The Wolfe conditions implemented in CCOMB and CG_DESCENT (HZw) can compute a solution with accuracy of the order of the square root of the machine epsilon.

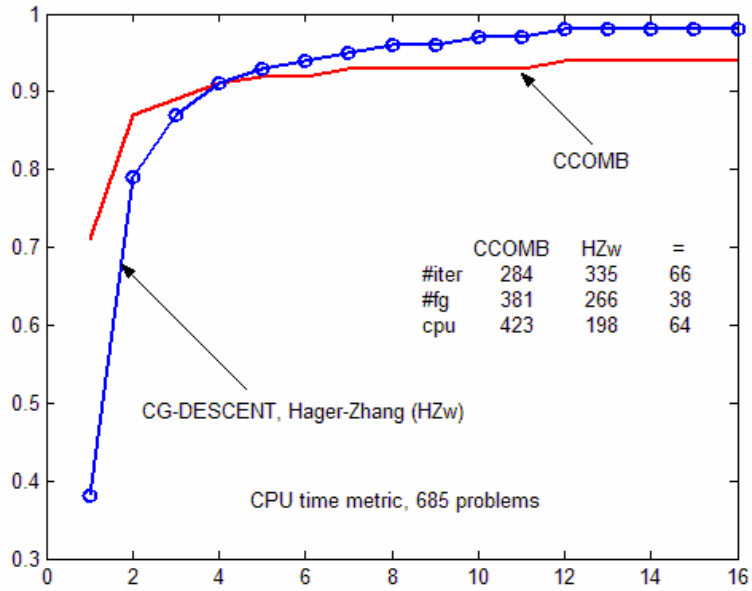

Fig. 10. Performance based on CPU time. CCOMB versus CG_DESCENT with Wolfe line search (HZw).

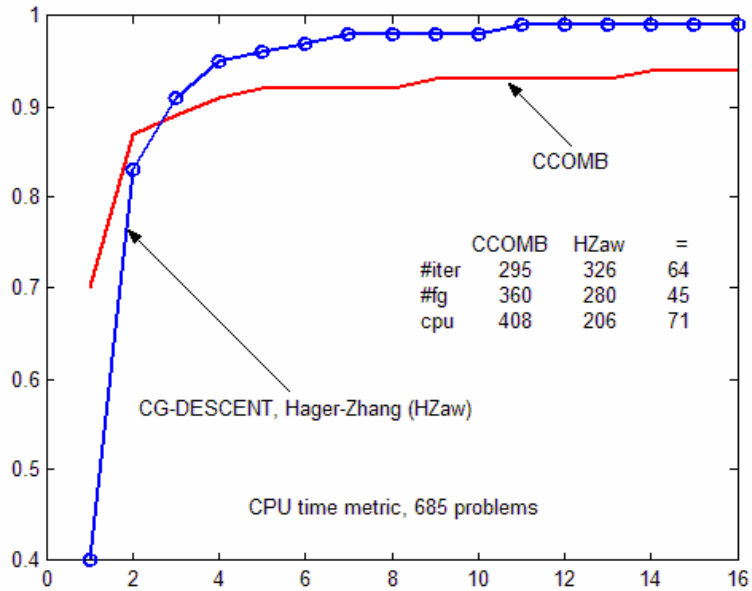

Fig. 11. Performance based on CPU time. CCOMB versus CG_DESCENT with approximate Wolfe line search (HZaw).

In contrast, the approximate Wolfe line search implemented in CG_DESCENT (HZaw) can compute the solution with accuracy of the order of machine epsilon. Figures 10 and 11 present the performance profile of these algorithms in comparison to CCOMB. We see that CG_DESCENT is more robust than CCOMB.

5. Conclusion

We know a large variety of conjugate gradient algorithms. The known hybrid conjugate gradient algorithms are based on projection of classical conjugate gradient algorithms. In this paper we have proposed new hybrid conjugate gradient algorithms in which the famous parameter β_k is computed as a convex combination of β_k^{PRP} and β_k^{DY}, i.e. $\beta_k = (1-\theta_k)\beta_k^{PRP} + \theta_k \beta_k^{DY}$. The parameter θ_k is computed in such a manner that the conjugacy condition is satisfied, or the corresponding direction in hybrid conjugate gradient algorithm is the Newton direction. For uniformly convex functions if the stepsize s_k approaches zero, the gradient is bounded in the sense that $\eta_1 \|s_k\|^2 \le \|g_k\|^2 \le \eta_2 \|s_{k-1}\|$ and the line search satisfy the strong Wolfe conditions, then our hybrid conjugate gradient algorithms are globally convergent. For general nonlinear functions if the parameter θ_k from β_k^N definition is bounded, i.e. $0 < \theta_k < 1$, then our hybrid conjugate gradient is globally convergent. The Dolan and Moré CPU performance profile of hybrid conjugate gradient algorithm based on conjugacy condition (CCOMB algorithm) is higher than the performance profile corresponding to the hybrid algorithm based to the Newton direction (NDOMB algorithm). The performance profile of CCOMB algorithm was higher than those of the well established conjugate gradient algorithms (hDY, hDYz, GN, HuS, TS and LS-CD) for a set consisting of 750 unconstrained optimization test problems, some of them from CUTE library. Additionally the proposed hybrid conjugate gradient algorithm CCOMB is more robust than the PRP and DY conjugate gradient algorithms. However, CCOMB algorithm is outperformed by CG_DESCENT.

References

[1] N. Andrei, "*Test functions for unconstrained optimization*". http://www.ici.ro/camo/neculai/SCALCG/evalfg.for

[2] N. Andrei, *Scaled conjugate gradient algorithms for unconstrained optimization*. Accepted: Computational Optimization and Applications, 2006.

[3] N. Andrei, *Scaled memoryless BFGS preconditioned conjugate gradient algorithm for unconstrained optimization.* Accepted: Optimization Methods and Software, 2006.

[4] N. Andrei, *A scaled BFGS preconditioned conjugate gradient algorithm for unconstrained optimization*. Accepted: Applied Mathematics Letters, 2006

[5] E. Birgin and J.M. Martínez, *A spectral conjugate gradient method for unconstrained optimization*, Applied Math. and Optimization, 43, pp.117-128, 2001.

[6] I. Bongartz, A.R. Conn, N.I.M. Gould and P.L. Toint, *CUTE: constrained and unconstrained testing environments*, ACM Trans. Math. Software, 21, pp.123-160, 1995.

[7] Y.H. Dai, *Analysis of conjugate gradient methods,* Ph.D. Thesis, Institute of Computational Mathematics and Scientific/Engineering Computing, Chinese Academy of Science, 1997.

[8] Y.H. Dai, *New properties of a nonlinear conjugate gradient method.* Numer. Math., 89 (2001), pp.83-98.

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

[10] Y.H. Dai, L.Z. Liao and Duan Li, *On restart procedures for the conjugate gradient method.* Numerical Algorithms 35 (2004), pp. 249-260.

[11] Y.H. Dai, Han, J.Y., Liu, G.H., Sun, D.F., Yin, .X. and Yuan, Y., *Convergence properties of nonlinear conjugate gradient methods*. SIAM Journal on Optimization 10 (1999), 348-358.
[12] 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.
[13] Y.H. Dai and Y. Yuan, *An efficient hybrid conjugate gradient method for unconstrained optimization*, Ann. Oper. Res., 103 (2001), pp. 33-47.
[14] E.D. Dolan and J.J. Moré, "*Benchmarking optimization software with performance profiles*", Math. Programming, 91 (2002), pp. 201-213.
[15] R. Fletcher, *Practical Methods of Optimization, vol. 1: Unconstrained Optimization*, John Wiley & Sons, New York, 1987.
[16] R. Fletcher and C. Reeves, *Function minimization by conjugate gradients*, Comput. J., 7 (1964), pp.149-154.
[17] J.C. Gilbert and J. Nocedal, *Global convergence properties of conjugate gradient methods for optimization*, SIAM J. Optim., 2 (1992), pp. 21-42.
[18] W.W. Hager and H. Zhang, "*A new conjugate gradient method with guaranteed descent and an efficient line search*", SIAM Journal on Optimization, 16 (2005) 170-192.
[19] W.W. Hager and H. Zhang, *A survey of nonlinear conjugate gradient methods*. Pacific journal of Optimization, 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] Y.F. Hu and C. Storey, *Global convergence result for conjugate gradient methods.* J. Optim. Theory Appl., 71 (1991), pp.399-405.
[22] Y. Liu, and C. Storey, *Efficient generalized conjugate gradient algorithms, Part 1: Theory.* JOTA, 69 (1991), pp.129-137.
[23] E. Polak and G. Ribière, *Note sur la convergence de directions conjuguée,* Rev. Francaise Informat Recherche Operationelle, 3e Année 16 (1969), pp.35-43.
[24] B.T. Polyak, *The conjugate gradient method in extreme problems*. USSR Comp. Math. Math. Phys., 9 (1969), pp.94-112.
[25] M.J.D. Powell, *Restart procedures of the conjugate gradient method*. Mathematical Programming, 2 (1977), pp.241-254.
[26] M.J.D. Powell, *Nonconvex minimization calculations and the conjugate gradient method.* in Numerical Analysis (Dundee, 1983), Lecture Notes in Mathematics, vol. 1066, Springer-Verlag, Berlin, 1984, pp.122-141.
[27] D.F. Shanno and K.H. Phua, *Algorithm 500, Minimization of unconstrained multivariate functions*, ACM Trans. on Math. Soft., 2, pp.87-94, 1976.
[28] D. Touati-Ahmed and C. Storey, *Efficient hybrid conjugate gradient techniques.* J. Optim. Theory Appl., 64 (1990), pp.379-397.
[29] Y.Yuan, *Analysis on the conjugate gradient method*, Optimization Methods and Software, 2 (1993), pp.19-29.

September 26, 2007
Sent to Panos Pardalos