# A hybrid conjugate gradient algorithm for unconstrained optimization as a convex combination of Hestenes-Stiefel and Dai-Yuan

**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.** In this paper we propose and analyze another hybrid conjugate gradient algorithm in which the parameter $\beta_k$ is computed as a convex combination of $\beta_k^{HS}$ (Hestenes-Stiefel) and $\beta_k^{DY}$ (Dai-Yuan), i.e. $\beta_k^C = (1-\theta_k)\beta_k^{HS} + \theta_k\beta_k^{DY}$. The parameter $\theta_k$ in the convex combination is computed in such a way that the direction corresponding to the conjugate gradient algorithm is the Newton direction and the secant equation is satisfied. 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 this hybrid computational scheme outperforms the Hestenes-Stiefel and the Dai-Yuan conjugate gradient algorithms as well as some other known hybrid conjugate gradient algorithms. Comparisons with CG_DESCENT by Hager and Zhang [17] and LBFGS by Liu and Nocedal [22] show that CG_DESCENT is more robust then our algorithm, and LBFGS is top performer among these algorithms.

*MSC*: 49M07, 49M10, 90C06, 65K
*Keywords*: Unconstrained optimization, hybrid conjugate gradient method, Newton direction, numerical comparisons

## 1. Introduction

In this paper 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. As we know, 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) $\beta_k$ 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 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 [18]. Different conjugate gradient algorithms correspond to different choices for the scalar parameter $\beta_k$. The methods of Fletcher and

Reeves (FR) [15], of Dai and Yuan (DY) [11] and the Conjugate Descent (CD) proposed by Fletcher [14]:

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

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], of Hestenes and Stiefel (HS) [19] or of Liu and Storey (LS) [21]:

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

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 and to improve the performances of the above conjugate gradient algorithms. One of the first hybrid conjugate gradient algorithms has been introduced by Touati-Ahmed and Storey [27], where the parameter $\beta_k$ 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 $\beta_k^{PRP}$ tends to zero. Therefore, $\beta_k^{PRP}$ 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 [20], where $\beta_k$ 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 [14] 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 [16] 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 $\beta_k^{FR}$ is always nonnegative, it follows that $\beta_k^{GN}$ 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 [12] combined in a projective manner their algorithm with that of Hestenes and Stiefel, thus 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 [12] established the global convergence of these hybrid computational schemes.

In contrast to the hybrid methods $\beta_k^{hDY}$ and $\beta_k^{hDYz}$ in this paper we propose another hybrid conjugate gradient where the parameter $\beta_k$ is computed as a *convex combination* of $\beta_k^{HS}$ and $\beta_k^{DY}$. We selected these two methods to combine in a hybrid conjugate gradient algorithm because HS has good computational properties, on one side, and DY has strong convergence properties, on the other side. HS method automatically adjust $\beta_k$ to avoid jamming, often this method performs better in practice than DY and we use this in order to have a good practical conjugate gradient 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 algorithm and in section 4 we show its convergence analysis. In section 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 [7], as well as some other unconstrained optimization problems presented in [1] show that this hybrid conjugate gradient algorithm outperforms the known hybrid conjugate gradient algorithms. However, the comparisons between our algorithm and CG_DESCENT by Hager and Zhang [17] show that CG_DESCENT is more robust. On the other hand, comparisons with LBFGS by Liu and Nocedal [22] show that limited memory LBFGS is top performer.

## 2. The hybrid conjugate gradient algorithm as a convex combination of HS and DY algorithms

Our algorithm generates the iterates $x_0, x_1, x_2, \ldots$ 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:

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

where

$$\beta_k^C = (1-\theta_k)\beta_k^{HS} + \theta_k\beta_k^{DY} = (1-\theta_k)\frac{g_{k+1}^T y_k}{y_k^T s_k} + \theta_k \frac{g_{k+1}^T g_{k+1}}{y_k^T s_k} \tag{7}$$

and $\theta_k$ is a scalar parameter satisfying $0 \le \theta_k \le 1$, which follows to be determined. Observe that if $\theta_k = 0$, then $\beta_k^C = \beta_k^{HS}$, and if $\theta_k = 1$, then $\beta_k^C = \beta_k^{DY}$. On the other hand, if $0 < \theta_k < 1$, then $\beta_k^C$ is a convex combination of $\beta_k^{HS}$ and $\beta_k^{DY}$.

The HS method has the property that the conjugacy condition $y_k^T d_{k+1} = 0$ always holds, independent of the line search. With an exact line search $\beta_k^{HS} = \beta_k^{PRP}$. Therefore, the convergence properties of the HS methods are similar to the convergence properties of the PRP method. As a consequence, by Powell's example [25], the HS method with the exact line search, for general nonlinear functions, may not converge. The HS method has a built-in restart feature that addresses directly to the jamming phenomenon. Indeed, when the step $x_{k+1} - x_k$ is small, then the factor $y_k = g_{k+1} - g_k$ in the numerator of $\beta_k^{HS}$ tends to zero.

Hence, $\beta_k^{HS}$ becomes small and the new direction $d_{k+1}$ is essentially the steepest descent direction $-g_{k+1}$. The performance of HS method is better than the performance of DY [5,18].

The DY method, on the other side, 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 $\gamma_1$ and $\gamma_2$ 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 $\le j$.

Clearly, from (6) and (7) it is easy to see that

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

In our algorithm the parameter $\theta_k$ is selected in such a manner that the direction $d_{k+1}$ given by (8) is equal with the Newton direction $d_{k+1}^N = -\nabla^2 f(x_{k+1})^{-1} g_{k+1}$. Therefore, from the equation

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

after some algebra, we get

$$\theta_k = \frac{s_k^T \nabla^2 f(x_{k+1}) g_{k+1} - s_k^T g_{k+1} - \dfrac{y_k^T g_{k+1}}{y_k^T s_k} s_k^T \nabla^2 f(x_{k+1}) s_k}{\left[ \dfrac{g_{k+1}^T g_{k+1}}{y_k^T s_k} - \dfrac{y_k^T g_{k+1}}{y_k^T s_k} \right] s_k^T \nabla^2 f(x_{k+1}) s_k}. \tag{9}$$

However, in this formula the salient point is the presence of the Hessian. For large-scale problems, choices for the update parameter that do not require the evaluation of the Hessian matrix are often preferred in practice to the methods that require the Hessian in each iteration. Therefore, in order to have an algorithm for solving large-scale problems we assume that the pair $(s_k, y_k)$ satisfies the secant equation $\nabla^2 f(x_{k+1}) s_k = y_k$. This leads us to:

$$\theta_k = -\frac{s_k^T g_{k+1}}{g_k^T g_{k+1}}. \tag{10}$$

Obviously, using (10) in (8) our direction can be expressed as:

$$d_{k+1} = -Q_{k+1} g_{k+1}, \tag{11}$$

where

$$Q_{k+1} = I - \left( 1 + \frac{s_k^T g_{k+1}}{g_k^T g_{k+1}} \right) \frac{s_k y_k^T}{y_k^T s_k} + \frac{s_k^T g_{k+1}}{g_k^T g_{k+1}} \frac{s_k g_{k+1}^T}{y_k^T s_k} \tag{12}$$

is another rank two approximation to the inverse of the Hessian. As known, the secant equation does not hold exactly in non-quadratic problems. Zhang *et al* [28] proved that if $\|s_k\|$ is sufficiently small, then $s_k^T \nabla^2 f(x_{k+1}) s_k - s_k^T y_k = O(\|s_k\|^3)$. Therefore, the direction (11) is an approximation of the Newton direction. A major difficulty with this approach is that the matrix $Q_{k+1}$ defined by (12) is not symmetric and hence not positive definite. Thus the corresponding directions are not necessarily descent and numerical instability can result. This is the price we must pay by using the secant equation in (9) to get (10). With exact line searches ( $s_k^T g_{k+1} = 0$ ), $d_{k+1} = -Q_{k+1} g_{k+1}$ reduces to the Hestenes and Stiefel method.

**Theorem 1.** *Assume that* $d_k$ *is a descent direction and* $\alpha_k$ *in algorithm (2) and (8), where* $\theta_k$ *is given by (10), is determined by the Wolfe line search (4) and (5). If* $0 < \theta_k < 1$, *and*

$$\frac{(y_k^T g_{k+1})(s_k^T g_{k+1})}{y_k^T s_k} \le \|g_{k+1}\|^2, \tag{13}$$

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

***Proof***. From (8) and (10) we get

$$g_{k+1}^T d_{k+1} = -\left[1 + \frac{(s_k^T g_{k+1})^2}{(g_k^T g_{k+1})(y_k^T s_k)}\right]\|g_{k+1}\|^2 + \frac{(y_k^T g_{k+1})(s_k^T g_{k+1})}{y_k^T s_k}\left[1 + \frac{s_k^T g_{k+1}}{g_k^T g_{k+1}}\right]. \tag{14}$$

Since $s_k^T g_k < 0$ it follows that $s_k^T g_{k+1} = y_k^T s_k + s_k^T g_k < y_k^T s_k$, i.e.

$$\frac{s_k^T g_{k+1}}{y_k^T s_k} < 1. \tag{15}$$

On the other hand, $0 < \theta_k < 1$, hence

$$0 < 1 + \frac{s_k^T g_{k+1}}{g_k^T g_{k+1}} < 1. \tag{16}$$

Therefore, using (13) we have

$$\begin{aligned} g_{k+1}^T d_{k+1} &\le -\left[1 + \frac{(s_k^T g_{k+1})^2}{(g_k^T g_{k+1})(y_k^T s_k)}\right]\|g_{k+1}\|^2 + \left[1 + \frac{s_k^T g_{k+1}}{g_k^T g_{k+1}}\right]\|g_{k+1}\|^2 \\ &= -\left(\frac{s_k^T g_{k+1}}{g_k^T g_{k+1}}\right)\left[\frac{s_k^T g_{k+1}}{y_k^T s_k} - 1\right]\|g_{k+1}\|^2 \le 0 \end{aligned} \tag{17}$$

proving that the direction $d_{k+1}$ is a descent one. ■

**Theorem 2.** *Assume that the conditions in Theorem 1 hold. If there exists a constant* $c_1 > 0$, *so that* $0 < c_1 \le \theta_k < 1$, *then there exists a constant* $\delta > 0$ *so that*

$$g_{k+1}^T d_{k+1} \le -\delta \|g_{k+1}\|^2, \tag{18}$$

*i.e. the direction* $d_{k+1}$ *satisfies the sufficient descent condition.*

**Proof.** From (17) we have

$$g_{k+1}^T d_{k+1} \le -\left(\frac{s_k^T g_{k+1}}{g_k^T g_{k+1}}\right)\left(\frac{s_k^T g_k}{y_k^T s_k}\right)\|g_{k+1}\|^2. \tag{19}$$

Since $y_k^T s_k > 0$ and $s_k^T g_k \le 0$, it follows that there exists a constant $c_2 > 0$, so that $g_k^T s_k \le -c_2 (y_k^T s_k) < 0$. On the other hand, since $1 > \theta_k \ge c_1 > 0$, then $s_k^T g_{k+1} \le -c_1 (g_k^T g_{k+1})$. Therefore, from (19) we have

$$g_{k+1}^T d_{k+1} \le -\left(\frac{s_k^T g_{k+1}}{g_k^T g_{k+1}}\right)\left(\frac{s_k^T g_k}{y_k^T s_k}\right)\|g_{k+1}\|^2 \le -c_1 c_2 \|g_{k+1}\|^2 \equiv -\delta \|g_{k+1}\|^2,$$

where $\delta = c_1 c_2 > 0$. ■

The parameter $\theta_k$ given by (10) 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. $\beta_k^C = \beta_k^{HS}$; if $\theta_k \ge 1$, then take $\theta_k = 1$ in (7), i.e. $\beta_k^C = \beta_k^{DY}$. Therefore,

under this rule for $\theta_k$ selection, the direction $d_{k+1}$ in (8) combines the HS and DY algorithms in a convex way.

## 3. The HYBRID algorithm

*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 $\alpha_0 = 1/\|g_0\|$.

*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 conditions (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. $\theta_k$ parameter computation*. If $g_k^T g_{k+1} = 0$, then set $\theta_k = 0$, otherwise compute $\theta_k$ as in (10).

*Step 5. $\beta_k^C$ conjugate gradient parameter computation*. If $0 < \theta_k < 1$, then compute $\beta_k^C$ as in (7). If $\theta_k \ge 1$, then set $\beta_k^C = \beta_k^{DY}$. If $\theta_k \le 0$, then set $\beta_k^C = \beta_k^{HS}$.

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

$$\left| g_{k+1}^T g_k \right| \ge 0.2 \|g_{k+1}\|^2 \tag{20}$$

is satisfied, then restart, i.e. 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 $\alpha_k$ 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 $-g_{k+1}$. Under reasonable assumptions, conditions (4), (5) and (20) 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 $k \ge 1$ the starting guess for the steplength $\alpha_k$ in the line search is computed as $\alpha_{k-1} \|d_{k-1}\|_2 / \|d_k\|_2$. This selection was used for the first time by Shanno and Phua in CONMIN [26]. It was also considered in the packages: SCG by Birgin and Martínez [6] and in SCALCG by Andrei [2,3,4].

## 4. Convergence analysis

Assume that:

*(i)* *The level set* $S = \{x \in R^n : f(x) \le f(x_0)\}$ *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* $\|\nabla f(x) - \nabla f(y)\| \le L \|x - y\|$, *for all* $x, y \in N$.

Under these assumptions on $f$ there exists a constant $\Gamma \ge 0$ such that $\|\nabla f(x)\| \le \Gamma$ for all $x \in S$.

In [10] it is proved that for any conjugate gradient method with strong Wolfe line search the following general result 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* $\alpha_k$ *is obtained by the strong Wolfe line search*

$$f(x_k + \alpha_k d_k) - f(x_k) \le \rho \alpha_k g_k^T d_k, \tag{21}$$

$$\left| g_{k+1}^T d_k \right| \le \sigma g_k^T d_k. \tag{22}$$

*If*

$$\sum_{k \ge 1} \frac{1}{\left\| d_k \right\|^2} = \infty, \tag{23}$$

*then*

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

For uniformly convex functions which satisfy the above assumptions we can prove that the norm of $d_{k+1}$ generated by (8) and (10) 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), (8) and (10), where* $d_{k+1}$ *is a descent direction and* $\alpha_k$ *is obtained by the strong Wolfe line search (21) and (22). If for* $k \ge 0$, $0 < \theta_k < 1$ *and there exists the nonnegative constant* $\eta_1$ *such that*

$$\left\| g_{k+1} \right\|^2 \le \eta_1 \left\| s_k \right\|, \tag{25}$$

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

$$(\nabla f(x) - \nabla f(y))^T (x - y) \ge \mu \left\| x - y \right\|^2, \tag{26}$$

*then*

$$\lim_{k \to \infty} g_k = 0. \tag{27}$$

***Proof***. From (26) it follows that $y_k^T s_k \ge \mu \left\| s_k \right\|^2$. Now, since $0 < \theta_k < 1$, from uniform convexity and (25) we have:

$$\left| \beta_k^C \right| \le \left| \frac{y_k^T g_{k+1}}{y_k^T s_k} \right| + \left| \frac{g_{k+1}^T g_{k+1}}{y_k^T s_k} \right| \le \frac{\left\| g_{k+1} \right\| \left\| y_k \right\|}{\mu \left\| s_k \right\|^2} + \frac{\eta_1 \left\| s_k \right\|}{\mu \left\| s_k \right\|^2}. \tag{28}$$

But $\left\| y_k \right\| \le L \left\| s_k \right\|$, therefore

$$\left| \beta_k^C \right| \le \frac{\Gamma L}{\mu \left\| s_k \right\|} + \frac{\eta_1}{\mu \left\| s_k \right\|}.$$

Hence, with (28) we have

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

which implies that (23) is true. Therefore, by Lemma 1 we have (24), which for uniformly convex functions is equivalent to (27). ■

For general nonlinear functions the convergence analysis of our algorithm exploits insights developed by Gilbert and Nocedal [16], Dai and Liao [9] and by Hager and Zhang [17]. Global convergence proof of the HYBRID 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 $S$ is bounded and the function $f$ is bounded from below. Additionally, assume that there exists a constant $\gamma \geq 0$, such that $\gamma \leq \|g_k\|$.

**Theorem 4**. *Suppose that the assumptions (i) and (ii) hold and for every* $k \geq 0$ *there exist the constants* $\eta \geq 0$ *and* $\omega \geq 0$ *such that:* $\|g_{k+1}\| \leq \eta \|s_k\|$ and $\|g_{k+1}\| \leq \omega \|g_k\|^2 / \|s_k\|^2$. *If* $d_k$ *is a descent direction and* $\nabla f(x)$ *is a Lipschitz function on* $S$, *then for the computational scheme (2), (8) and (10), where* $0 < c_1 \leq \theta_k < 1$ *and* $\alpha_k$ *determined by the Wolfe line search (4)-(5) is bounded, either* $g_k = 0$ *for some* $k$ *or*

$$\liminf_{k \to \infty} \|g_k\| = 0. \tag{29}$$

***Proof.*** Since $0 < \theta_k < 1$ we can write

$$\left|\beta_k^C\right| \leq \left|\frac{y_k^T g_{k+1}}{y_k^T s_k}\right| + \left|\frac{g_{k+1}^T g_{k+1}}{y_k^T s_k}\right| \leq \frac{\|g_{k+1}\|}{\left|y_k^T s_k\right|}\left[\|y_k\| + \|g_{k+1}\|\right]. \tag{30}$$

By the Wolfe condition (5) we have:

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

On the other hand, since $0 < c_1 \leq \theta_k < 1$, then from theorem 2 there exists the constant $\delta > 0$ such that, $g_k^T s_k \leq -\delta \|g_k\|^2$. Therefore, $y_k^T s_k \geq (1-\sigma)\delta \|g_k\|^2$. Hence,

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

On the other hand, from Lipschitz continuity we have $\|y_k\| = \|g_{k+1} - g_k\| \leq L\|s_k\|$.
With these, from (30) we get

$$\left|\beta_k^C\right| \leq \frac{\omega}{(1-\sigma)\delta} \frac{1}{\|s_k\|^2}\left[L\|s_k\| + \eta\|s_k\|\right] = \frac{\omega(L+\eta)}{(1-\sigma)\delta} \frac{1}{\|s_k\|}. \tag{31}$$

Now, we can write

$$\|d_{k+1}\| \leq \|g_{k+1}\| + \left|\beta_k^C\right| \|s_k\| \leq \Gamma + \frac{\omega(L+\eta)}{(1-\sigma)\delta}. \tag{32}$$

Since the level set $S$ is bounded and the function $f$ is bounded from below, from (4) it follows that

$$0 < \sum_{k=0}^{\infty} \frac{(g_k^T d_k)^2}{\|d_k\|^2} < +\infty , \tag{33}$$

i.e. the Zoutendijk condition holds. Therefore, the descent property $g_k^T s_k \leq -\delta \|g_k\|^2$ yields:

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

which contradicts (32). Hence, $\gamma = \liminf_{k \to \infty} \|g_k\| = 0$. ■

## 5. Numerical experiments

In this section we present the computational performance of a Fortran implementation of the HYBRID algorithm on a set of 750 unconstrained optimization test problems. The test problems are the unconstrained problems in the CUTE [7] 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 [13]. All algorithms implement the Wolfe line search conditions with $\rho = 0.0001$ and $\sigma = 0.9$. The same stopping criterion $\|g_k\|_\infty \leq 10^{-6}$ is used, 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:

$$\left| f_i^{ALG1} - f_i^{ALG2} \right| < 10^{-3} \tag{34}$$

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. In this numerical study we declare that a method solved a particular problem if the final point obtained has the lowest functional value among the tested methods (up to $10^{-3}$ tolerance as it was specified in (34)). Clearly, this criterion is acceptable for users that are interested in minimizing functions and not finding critical points.

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.

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

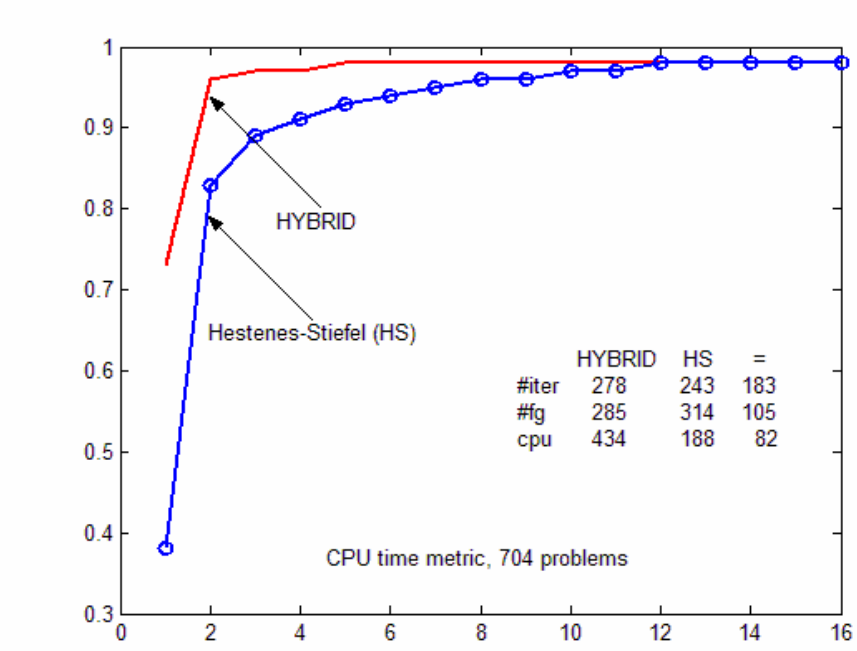


**Fig. 1.** Performance based on CPU time. HYBRID versus Hestenes and Stiefel (HS).

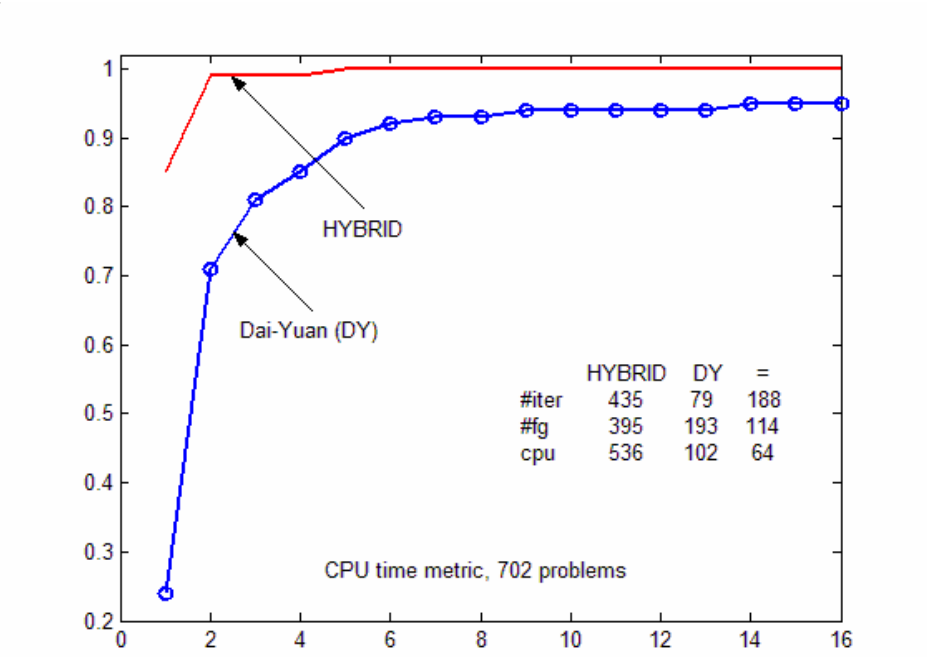


**Fig. 2.** Performance based on CPU time. HYBRID versus Dai-Yuan (DY).

When comparing HYBRID to HS (Figure 1), subject to the number of iterations, we see that HYBRID was better in 278 problems (i.e. it achieved the minimum number of iterations in 278 problems), HS was better in 243 problems and they achieved the same number of iterations in 183 problems, etc. Out of 750 problems, only for 704 problems does the criterion (34) hold. Similarly, in Figure 2 we see the number of problems for which HYBRID was better than DY. The second set of numerical experiments refers to the comparisons of HYBRID with other known hybrid conjugate gradient algorithms: hDY, hDYZ, GN, HuS, TS and LS-CD. Figures 3-8 present the Dolan and Moré CPU performance profiles of these algorithms, as well as the number of problems solved by each of these algorithms in minimum number of iterations, minimum number of function evaluations and minimum CPU time, respectively.

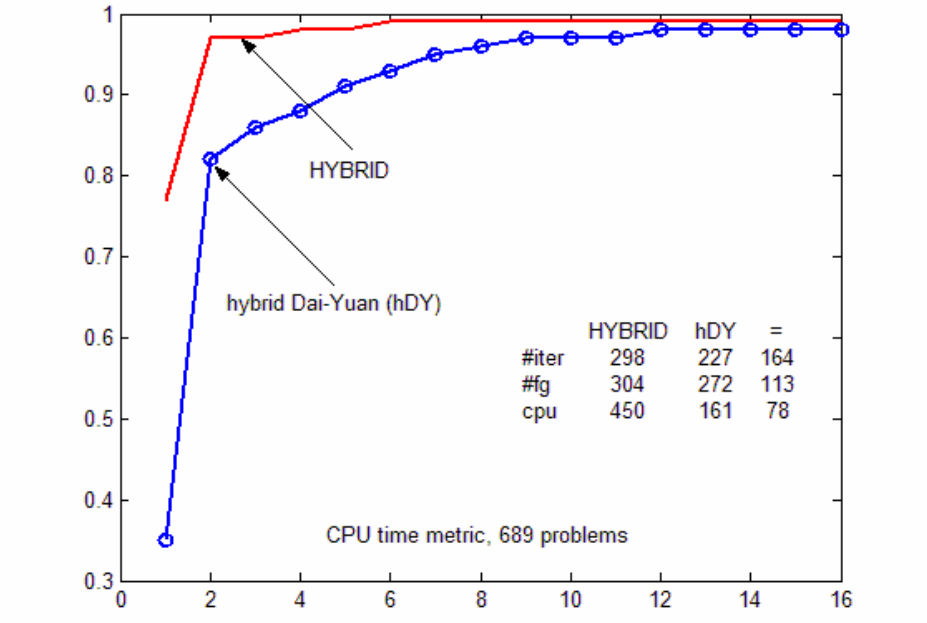


**Fig. 3.** Performance based on CPU time. HYBRID versus hybrid Dai-Yuan (hDY).

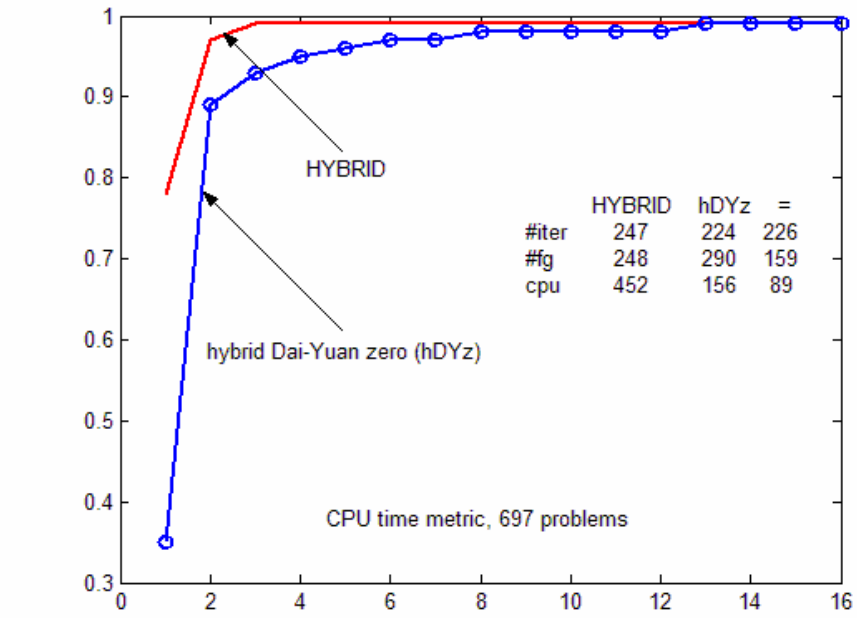


**Fig. 4.** Performance based on CPU time. HYBRID versus hybrid Dai-Yuan (hDYz).

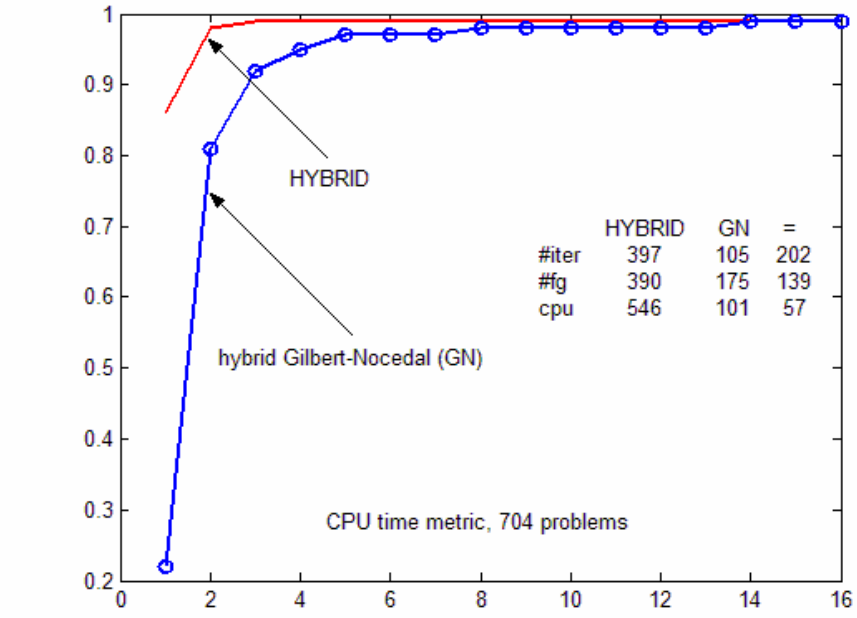


**Fig. 5.** Performance based on CPU time. HYBRID versus Gilbert-Nocedal (GN).

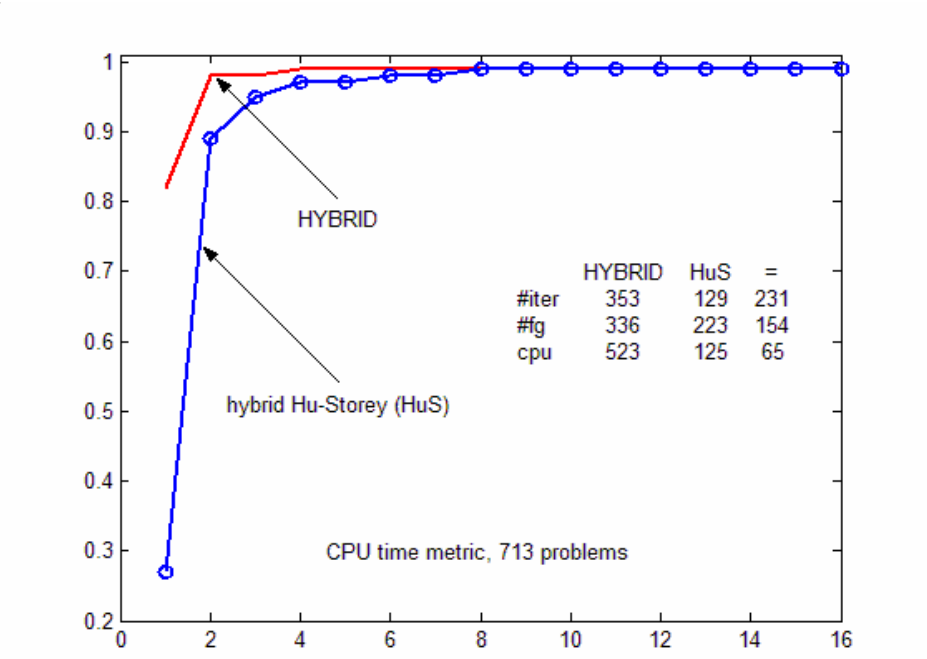


**Fig. 6.** Performance based on CPU time. HYBRID versus Hu-Storey (HuS).

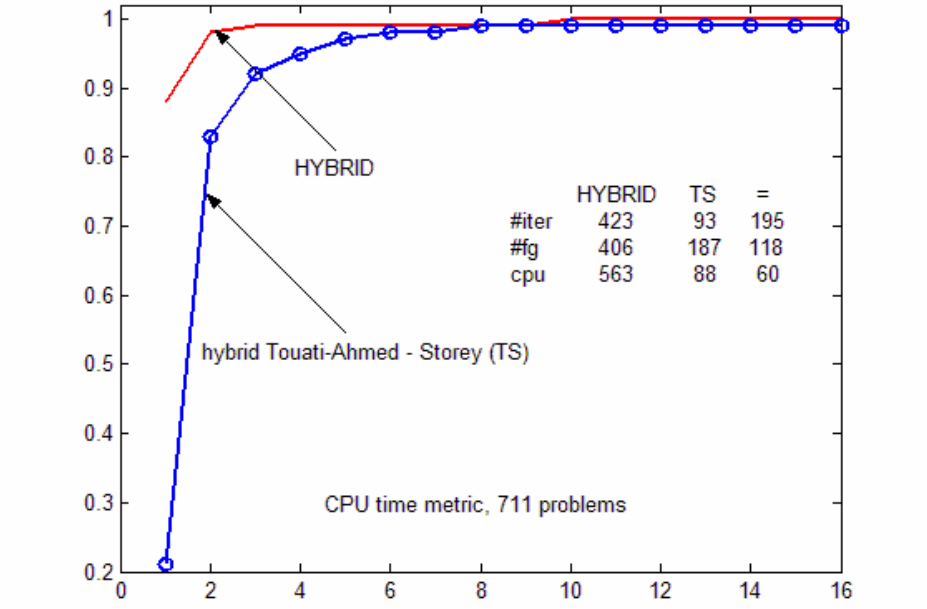


**Fig. 7.** Performance based on CPU time. HYBRID versus Touati-Ahmed - Storey (TS).

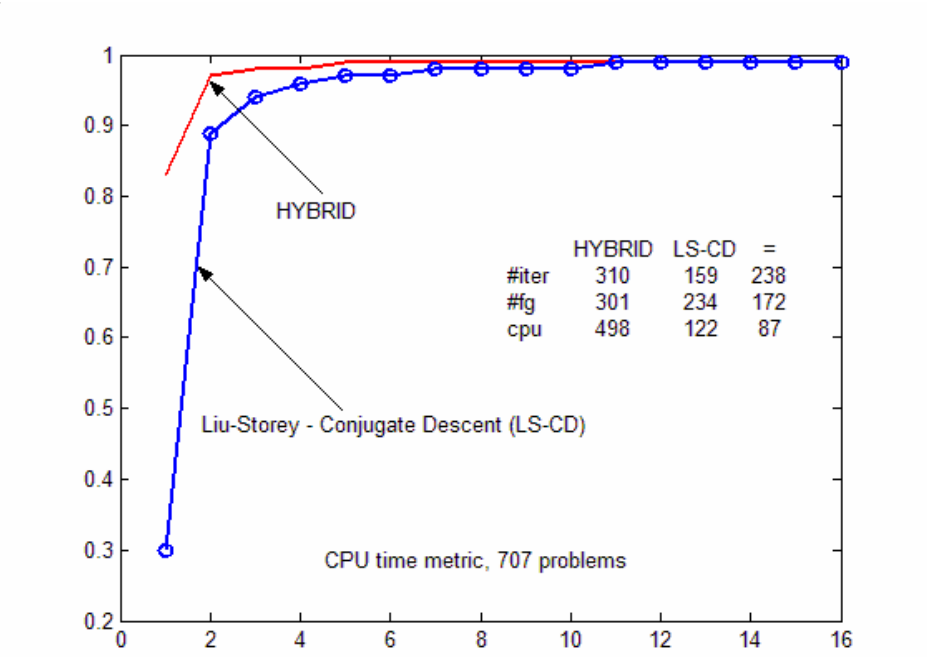


**Fig. 8.** Performance based on CPU time. HYBRID versus Liu-Storey – Conjugate Descent (LS-CD).

From the above Figures we see that HYBRID 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 hybrid conjugate gradient algorithms we considered here, HYBRID appears to generate the best search direction.

In the third set of numerical experiments we compare HYBRID to the CG_DESCENT conjugate gradient algorithm of Hager and Zhang [17]. 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.

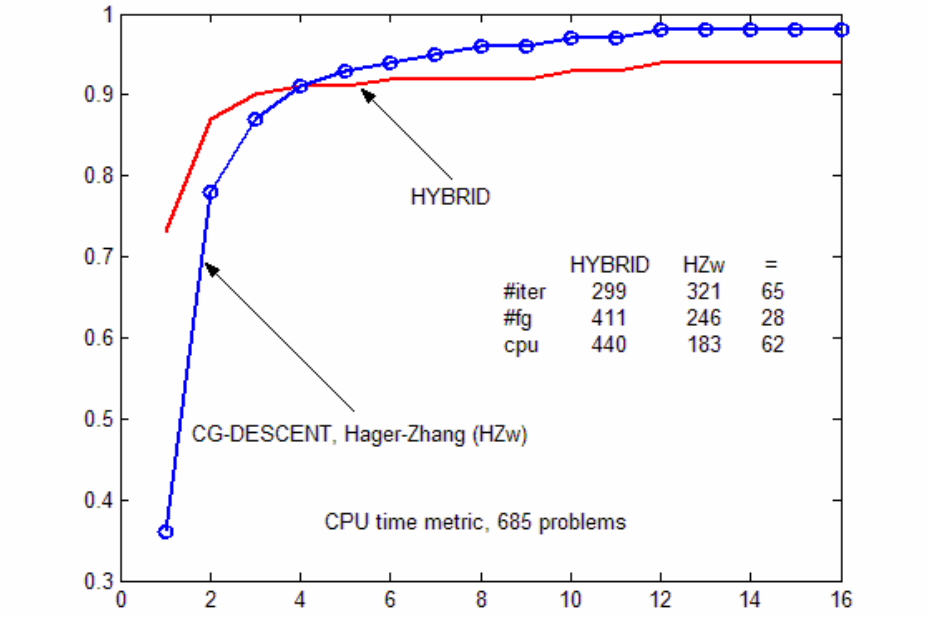


**Fig. 9.** Performance based on CPU time. HYBRID versus CG_DESCENT with Wolfe line search (HZw).

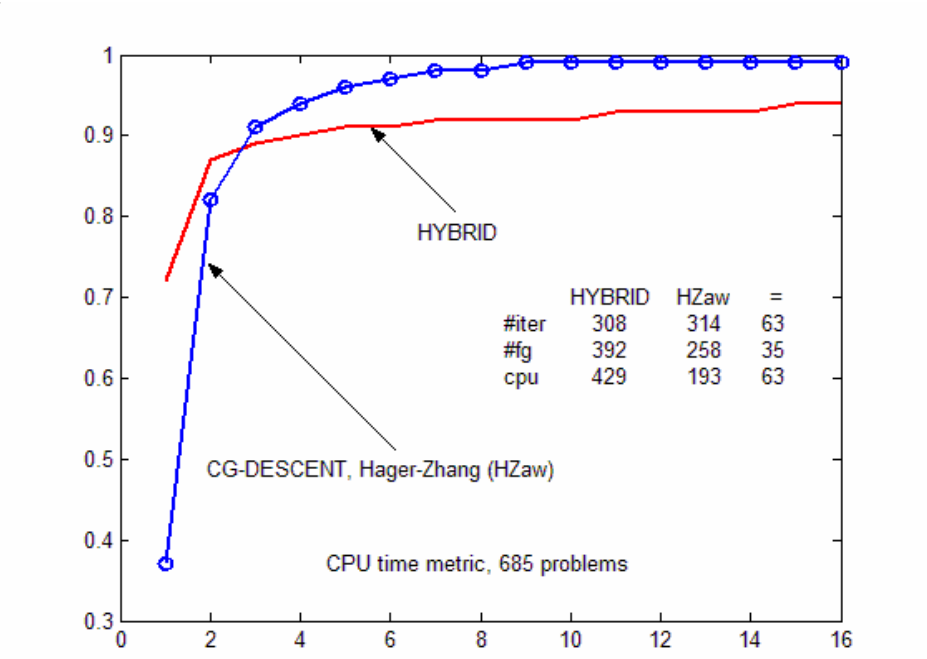


**Fig. 10.** Performance based on CPU time. HYBRID versus CG_DESCENT with approximate Wolfe line search (HZaw).

Figures 9 and 10 present the performance profile of these algorithms in comparison to HYBRID. We see that CG_DESCENT is more robust than HYBRID.

Finally, we compare our HYBRID conjugate gradient algorithm with limited memory LBFGS, LBFGS (m=3) of Liu and Nocedal [22]. Figure 11 presents the performance profile of these algorithms. We see that LBFGS is way more efficient than HYBRID.

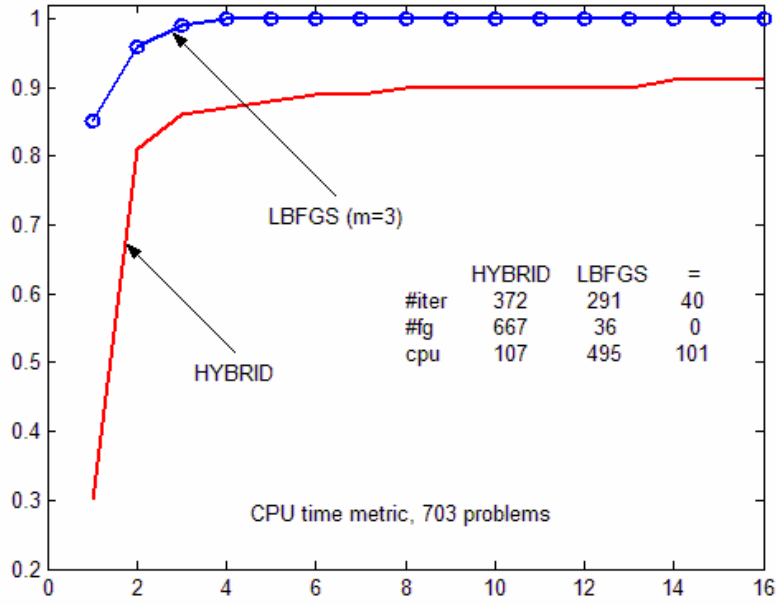


**Fig. 11.** Performance based on CPU time. HYBRID versus LBFGS (m=3).

Among the hybrid conjugate gradient algorithms our HYBRID algorithm is top performer. Also, the algorithm has better performance profiles than those corresponding to HS and DY. In this numerical study we noticed that for most of the iterations the HYBRID algorithm

uses $\beta_k^C$. Referring to the condition (13) we noticed that $(y_k^T g_{k+1})(s_k^T g_{k+1})/y_k^T s_k$ tends to zero faster than $\|g_{k+1}\|^2$. For most of the iterations the condition (13) is satisfied, i.e. the algorithm has a self-adjusting property in the sense given in [8]. It is worth saying that the condition (13) is more satisfied after those iterations in which $\beta_k^C$ is computed according to the HS or DY rules. Introducing (13) as a restart criterion, does not improve the performances of the algorithm. On the other hand, the conditions $\|g_{k+1}\| \le \eta \|s_k\|$ and $\|g_{k+1}\| \le \omega \|g_k\|^2 / \|s_k\|^2$ from theorem 4 say that $\|g_{k+1}\|^3 \le \omega\eta^2 \|g_k\|^2$. We noticed that there exists a $k_0$ such that for any iteration $k \ge k_0$ the above condition $\|g_{k+1}\|^3 \le \omega\eta^2 \|g_k\|^2$ is satisfied, thus illustrating the global convergence of the algorithm.

## 5. Conclusion

We know a large variety of conjugate gradient algorithms. In this paper we have presented a new hybrid conjugate gradient algorithm in which the famous parameter $\beta_k$ is computed as a convex combination of $\beta_k^{HS}$ and $\beta_k^{DY}$. For uniformly convex functions if the gradient is bounded in the sense that $\|g_k\|^2 \le \eta_1 \|s_{k-1}\|$ and the line search satisfies the strong Wolfe conditions then our hybrid conjugate gradient algorithm is globally convergent. For general nonlinear functions if the parameter $\theta_k$ from definition of $\beta_k^C$ is bounded, and both $\|g_{k+1}\| \le \eta \|s_k\|$ and $\|g_{k+1}\| \le \omega \|g_k\|^2 / \|s_k\|^2$ are satisfied, where $\eta$ and $\omega$ are nonnegative constants, then our hybrid conjugate gradient is globally convergent. The performance profile of our algorithm was higher than those of the well established hybrid conjugate gradient algorithms for a set consisting of 750 unconstrained optimization problems some of them from CUTE library and some others we presented in [1]. Additionally the proposed hybrid conjugate gradient algorithm is more robust than the HS and DY conjugate gradient algorithms. Concerning the robustness, the HYBRID algorithm is outperformed by CG_DESCENT. However, both HYBRID and CG_DESCENT are outperformed by limited memory LBFGS by Liu and Nocedal.

Finally, we remark that introducing different approximations of the Hessian matrix in (9) we get new hybrid conjugate gradient algorithms. One possibility is to consider the inverse of the spectral gradient choice: $y_k^T s_k / s_k^T s_k$. Another one is to use the relation: $s_k^T \nabla^2 f(x_k) s_k = s_k^T y_k + 6(f(x_k) - f(x_{k+1})) + 3(g_k + g_{k+1})^T s_k + O(\|s_k\|^4)$, which introduce the modified secant condition [29]. These approaches should be considered in another paper.

### References


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

[2] N. Andrei, *Scaled conjugate gradient algorithms for unconstrained optimization*. Computational Optimization and Applications, 38 (2007), pp. 401-416.

[3] N. Andrei, *Scaled memoryless BFGS preconditioned conjugate gradient algorithm for unconstrained optimization.* Optimization Methods and Software, 22 (2007), 561-571.

[4] N. Andrei, *A scaled BFGS preconditioned conjugate gradient algorithm for unconstrained optimization*. Applied Mathematics Letters, 20 (2007), 645-650.

[5] N. Andrei, *Numerical comparison of conjugate gradient algorithms for unconstrained optimization.* Studies in Informatics and Control, 16 (2007), pp.333-352.

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

[7] 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.
[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, Han, J.Y., Liu, G.H., Sun, D.F., Yin, .X. and Yuan, Y., *Convergence properties of nonlinear conjugate gradient methods*. SIAM Jurnal on Optimization 10 (1999), 348-358.
[11] 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.
[12] Y.H. Dai and Y. Yuan, *An efficient hybrid conjugate gradient method for unconstrained optimization*, Ann. Oper. Res., 103 (2001), pp. 33-47.
[13] E.D. Dolan and J.J. Moré, "*Benchmarking optimization software with performance profiles*", Math. Programming, 91 (2002), pp. 201-213.
[14] R. Fletcher, *Practical Methods of Optimization, vol. 1: Unconstrained Optimization*, John Wiley & Sons, New York, 1987.
[15] R. Fletcher and C. Reeves, *Function minimization by conjugate gradients*, Comput. J., 7 (1964), pp.149-154.
[16] J.C. Gilbert and J. Nocedal, *Global convergence properties of conjugate gradient methods for optimization*, SIAM J. Optim., 2 (1992), pp. 21-42.
[17] 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.
[18] W.W. Hager and H. Zhang, *A survey of nonlinear conjugate gradient methods*. Pacific journal of Optimization, 2 (2006), pp.35-58.
[19] 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.
[20] Y.F. Hu and C. Storey, *Global convergence result for conjugate gradient methods.* J. Optim. Theory Appl., 71 (1991), pp.399-405.
[21] Y. Liu, and C. Storey, *Efficient generalized conjugate gradient algorithms, Part 1: Theory.* JOTA, 69 (1991), pp.129-137.
[22] D.C. Liu, J. Nocedal, *On the limited memory BFGS method for large scale optimization.* Mathematical Programming, vol.45 (1989), pp.503-528.
[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, *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.
[26] D.F. Shanno and K.H. Phua, *Algorithm 500, Minimization of unconstrained multivariate functions*, ACM Trans. on Math. Soft., 2, pp.87-94, 1976.
[27] D. Touati-Ahmed and C. Storey, *Efficient hybrid conjugate gradient techniques.* J. Optim. Theory Appl., 64 (1990), pp.379-397.
[28] J.Z. Zhang, N.Y. Deng and L.H. Chen, *New quasi-Newton equation and related methods for unconstrained optimization.* J. Optim. Theory Appl., 102 (1999), pp.147-167.
[29] H. Yabe, and M. Takano, *Global convergence properties of nonlinear conjugate gradient methods with modified secant condition.* Computational Optimization and Applications, 28 (2004), pp.203-225.

**January 14, 2008**
**Sent to Studies in Informatics and Control**