A Hybrid Conjugate Gradient Algorithm with Modified Secant Condition for Unconstrained Optimization as a Convex Combination of Hestenes-Stiefel and Dai-Yuan Algorithms

Neculai Andrei

National Institute for R&D in Informatics, (ICI)

Center for Advanced Modeling and Optimization,

8-10, Averescu Avenue, Bucharest 1, Romania,

Academy of Romanian Scientists

54, Splaiul Independenței, Bucharest 5, Romania

E-mail: nandrei@ici.ro

Abstract: Another hybrid conjugate gradient algorithm is suggested in this paper. The parameter β_k is computed as a convex combination of β_k^{HS} (Hestenes-Stiefel) and β_k^{DY} (Dai-Yuan) formulae, i.e. $\beta_k^C = (1-\theta_k)\beta_k^{HS} + \theta_k \beta_k^{DY}$. The parameter θ_k in the convex combination is computed in such a way so that the direction corresponding to the conjugate gradient algorithm to be the Newton direction and the pair (s_k, y_k) to satisfy the modified secant condition given by Zhang *et al.* [32] and Zhang and Xu [33], where $s_k = x_{k+1} - x_k$ and $y_k = g_{k+1} - g_k$. The algorithm uses the standard Wolfe line search conditions. Numerical comparisons with conjugate gradient algorithms show that this hybrid computational scheme outperforms a variant of the hybrid conjugate gradient algorithm given by Andrei [6], in which the pair (s_k, y_k) satisfies the secant condition $\nabla^2 f(x_{k+1}) s_k = y_k$, as well as the Hestenes-Stiefel, the Dai-Yuan conjugate gradient algorithms, and the hybrid conjugate gradient algorithms of Dai and Yuan. A set of 750 unconstrained optimization problems are used, some of them from the CUTE library.

MSC: 49M07, 49M10, 90C06, 65K

Keywords: Unconstrained optimization, hybrid conjugate gradient method, Newton direction, numerical comparisons.

Neculai Andrei graduated in Mathematics, "A.I. Cuza" University – Iasi, and in Electrical Engineering from "Politehnica" University – Bucharest. He took his PhD degree for his contributions to Digraph Approach of Large-Scale Sparse Linear Dynamic Systems in 1984. He has been a senior scientific researcher at National Institute for R&D in Informatics - ICI, since 1973. He is author of 13 books and text books, as well as over 50 published papers and 300 Technical Reports in area of mathematical modeling optimization. He is the key architect of *ALLO language and compiler* for modeling in mathematical programming, as well as of some professional packages for linear programming and large-scale unconstrained optimization. His main current scientific interests centre on modeling for mathematical programming and on large-scale non-linear optimization, conjugate gradient, interior point methods, penalty and barrier approaches. Dr. Andrei is a founding member of *the Operations Research Society* of Romania; founding member of the *Center for Advanced Modeling and Optimization*; member of the Editorial Board of *Computational Optimization and Applications* – Springer Verlag (since 1992); member of *Computational Optimization and Applications – Optimization Forum*; member of the Editorial Board of *Journal of Systems Science and Complexity* – Science Press – China and Allerton Press – USA (since 2001). He is Editor-in-Chief to the *Advanced Modeling and Optimization – An Electronic International Journal* – ICI Press (since 1999). Dr. Andrei is an Alexander-von- Humboldt fellow, member ERCIM and of Academy of Romanian Scientists (since 2001).

1. Introduction

Let us consider the nonlinear unconstrained optimization problem

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

where $f: R^n \rightarrow R$ 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

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

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:

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

$$g(x_k + \alpha_k d_k)^T d_k \ge \sigma g_k^T d_k, \tag{5}$$

where d_k is a descent direction and $0 < \rho \le \sigma < 1$. Different conjugate gradient algorithms correspond to different choices for the scalar parameter β_k. The methods of Fletcher and Reeves (FR) [19], of Dai and Yuan (DY) [13] and the Conjugate Descent (CD) proposed by Fletcher [18]:

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

have strong convergence properties, but they may have modest practical performance due to jamming. On the other hand, the methods of Polak – Ribière [27] and Polyak (PRP) [28], of Hestenes and Stiefel (HS) [23] or of Liu and Storey (LS) [25]:

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

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

In this paper we focus on hybrid conjugate gradient methods. These algorithms have been devised to use the attractive features of the above conjugate gradient algorithms. They are defined by (2) and (3) where the parameter β_k is computed as projections or as convex combinations of different conjugate gradient algorithms, as in Table 1.

Table 1. Hybrid conjugate gradient algorithms.

Nr.	Formula	Author(s)
1.	$\beta_k^{hDY} = max\left\{c\beta_k^{DY}, min\left\{\beta_k^{HS}, \beta_k^{DY}\right\}\right\}$, $c = (1-\sigma)/(1+\sigma)$	Hybrid Dai-Yuan [14] (hDY)
2.	$\beta_k^{hDYz} = max\left\{0, min\left\{\beta_k^{HS}, \beta_k^{DY}\right\}\right\}$	Hybrid Dai-Yuan zero [14] (hDYz)
3.	$\beta_k^{GN} = max\left\{-\beta_k^{FR}, min\left\{\beta_k^{PRP}, \beta_k^{FR}\right\}\right\}$	Gilbert and Nocedal [20] (GN)
4.	$\beta_k^{HuS} = max\left\{0, min\left\{\beta_k^{PRP}, \beta_k^{FR}\right\}\right\}$	Hu and Storey [24] (HuS)
5.	$\beta_k^{TaS} = \begin{cases} \beta_k^{PRP} & 0 \le \beta_k^{PRP} \le \beta_k^{FR}, \\ \beta_k^{FR} & \text{otherwise} \end{cases}$	Touati-Ahmed and Storey [31] (TaS)

6.	$\beta_k^{LS-CD} = max\left\{0, min\left\{\beta_k^{LS}, \beta_k^{CD}\right\}\right\}$	Hybrid Liu-Storey, Conjugate-Descent (LS-CD)
7.	$\beta_k^{C} = (1-\theta_k)\beta_k^{HS} + \theta_k\beta_k^{DY}$, $0 < \theta_k < 1$, $\theta_k = -\frac{s_k^T g_{k+1}}{g_k^T g_{k+1}}$	Andrei [6] Newton direction. Secant condition.
8.	$\beta_k^{AC} = (1-\theta_k)\beta_k^{PRP} + \theta_k\beta_k^{DY}$, $0 < \theta_k < 1$, $\theta_k = \frac{(y_k^T g_{k+1})(y_k^T s_k) - (y_k^T g_{k+1})(g_k^T g_k)}{(y_k^T g_{k+1})(y_k^T s_k) - (g_{k+1}^T g_{k+1})(g_k^T g_k)}$.	Andrei [7] Conjugacy condition
9.	$\beta_k^{AN} = (1-\theta_k)\beta_k^{PRP} + \theta_k\beta_k^{DY}$, $0 < \theta_k < 1$, $\theta_k = \frac{(y_k^T g_{k+1} - s_k^T g_{k+1})\left\|g_k\right\|^2 - (g_{k+1}^T y_k)(y_k^T s_k)}{\left\|g_{k+1}\right\|^2 \left\|g_k\right\|^2 - (g_{k+1}^T y_k)(y_k^T s_k)}$.	Andrei [8] Newton direction

The hybrid computational schemes perform better than the classical conjugate gradient algorithms [5]. In [6] we presented another hybrid conjugate gradient algorithm as a convex combination of the Hestenes-Stiefel and the Dai-Yuan algorithms, where the parameter in convex combination is computed in such a way so that the direction corresponding to the conjugate gradient algorithm to be the Newton direction and the pair (s_k, y_k) to satisfy the secant condition. Numerical experiments with this computational scheme proved to outperform the Hestenes-Stiefel and the Dai-Yuan conjugate gradient algorithms, as well as some other hybrid conjugate gradient algorithms [6]. In this paper, motivated by a result given by Zhang, Deng and Chen [32] and Zhang and Xu [33] concerning a better approximation of $s_k^T \nabla^2 f(x_{k+1}) s_k$ using the modified secant condition, we present another variant of the hybrid conjugate gradient algorithm for unconstrained optimization which performs much better and it is more robust than the variant using the secant condition.

The structure of the paper is as follows. Section 2 introduces our hybrid conjugate gradient algorithm, HYBRIDM as a convex combination of HS and DY algorithms with modified secant condition. Section 3 presents the algorithm and in section 4 its convergence analysis both for uniformly convex functions and general functions is shown. In section 5 some numerical experiments and performance profiles of Dolan-Moré [17] corresponding to this new hybrid conjugate gradient algorithm are given. The performance profiles correspond to a set of 750 unconstrained optimization problems in the CUTE test problem library [10] as well as some other ones presented in [1]. It is shown that this hybrid conjugate gradient algorithm outperforms the classical HS and DY conjugate gradient algorithms and also the hybrid variants hDY and hDYz.

2. A Hybrid Conjugate Gradient Algorithm as a Convex Combination of HS and DY Algorithms with Modified Secant Condition

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 line search 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 θ_k is a scalar parameter satisfying $0 \le \theta_k \le 1$ which is 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 β_k^C is a convex combination of β_k^{HS} and β_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 [29], the HS method with an exact line search may not converge for general nonlinear functions. 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 β_k^{HS} tends to zero. Hence, β_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, 22].

On the other hand, the DY method always generates descent directions, and in [11] 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, we combine these two methods in a convex combination manner in order to have a good algorithm for unconstrained optimization. From (6) and (7) it is obvious that

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

Our motivation is to choose the parameter θ_k in such a way so that the direction d_{k+1} given by (8) to be the Newton direction. This is motivated by the fact that when the initial point x_0 is near the solution of (1) and the Hessian is a nonsingular matrix then the Newton direction is the best line search direction. Therefore, from the equality

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

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. One of the first conjugate gradient algorithm using the Hessian in the formula for β_k was given by Daniel [16] as $\beta_k = (g_{k+1}^T \nabla^2 f(x_k) d_k)/(d_k^T \nabla^2 f(x_k) d_k)$. For large-scale problems, choices for the update parameter β_k that do not require the evaluation of the Hessian matrix are often preferred in practice to the methods that require the Hessian at each iteration.

As we know, for quasi-Newton methods an approximation matrix B_k to the Hessian $\nabla^2 f(x_k)$ is used and updated so that the new matrix B_{k+1} satisfies the secant condition $B_{k+1}s_k = y_k$. Therefore, in order to have an algorithm for solving large-scale problems in [6] it is assumed that the pair (s_k, y_k) satisfies the secant condition. This leads us to a hybrid conjugate gradient algorithm (called HYBRID in [6]), where:

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

Observe that the secant condition does not hold exactly in non-quadratic problems. However, Zhang, Deng and Chen [32] 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 (8) and (10), where $0 < \theta_k < 1$, is an approximation of the Newton direction. Now, if $0 < \theta_k < 1$, then the direction given by (8) and (10) can be expressed as:

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

where

$$Q_{k+1} = I - \frac{s_k y_k^T}{y_k^T s_k} + \frac{s_k s_k^T}{y_k^T s_k}. \tag{12}$$

is a rank two approximation to the inverse of the Hessian. It is worth saying that the matrix Q_{k+1} in (12) was first proposed by Perry [26]. He arrived to this matrix by adding a correction term to the matrix modifying g_{k+1} in the direction corresponding to the HS method. 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 for 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.

Further, Zhang, Deng and Chen [32] and Zhang and Xu [33] expanded the secant condition and obtained a class of modified secant condition with a vector parameter which uses both the gradients and the function values in two successive points as:

$$B_{k+1}s_k = \hat{y}_k, \quad \hat{y}_k = y_k + \frac{\eta_k}{s_k^T u_k} u_k, \tag{13}$$

where $\eta_k = 6(f_k - f_{k+1}) + 3(g_k + g_{k+1})^T s_k$ and $u_k \in R^n$ is any vector such that $s_k^T u_k \neq 0$. Obviously, from (13) we get

$$s_k^T B_{k+1} s_k = s_k^T y_k + \eta_k. \tag{14}$$

Zhang, Deng and Chen [32] proved that if $\|s_k\|$ is sufficiently small, then for any vector u_k with $s_k^T u_k \neq 0$, $s_k^T \nabla^2 f(x_{k+1}) s_k - s_k^T \hat{y}_k = O(\|s_k\|^4)$ holds. Therefore, the quantity $s_k^T \hat{y}_k$ given by the modified secant condition (13) approximates the second-order curvature $s_k^T \nabla^2 f(x_{k+1}) s_k$ with a higher precision than the quantity $s_k^T y_k$ does. This is a very good motivation to use it in (9). For this purpose, in order to unify both approaches, we consider a slight modification of the modified secant condition (13) as $B_{k+1}s_k = z_k$, where

$$z_k = y_k + \frac{\delta\eta_k}{s_k^T u_k} u_k$$

and $\delta \ge 0$ is a scalar parameter. With $u_k = s_k$ this leads us to another hybrid conjugate gradient algorithm (2), (6) and (7), where

$$\theta_k = \frac{\left(\frac{\delta\eta_k}{s_k^T s_k} - 1\right) s_k^T g_{k+1} - \frac{y_k^T g_{k+1}}{y_k^T s_k}\delta\eta_k}{g_k^T g_{k+1} + \frac{g_k^T g_{k+1}}{y_k^T s_k}\delta\eta_k}. \tag{15}$$

Therefore, the direction (8) and (15), where $0 < \theta_k < 1$, is a better approximation of the Newton direction than that given by using (10) in (9). As above, observe that if $0 < \theta_k < 1$, then our direction can be expressed as:

$$d_{k+1} = -\bar{Q}_{k+1} g_{k+1}, \tag{16}$$

where

$$\bar{Q}_{k+1} = I - \frac{s_k y_k^T}{y_k^T s_k + \delta\eta_k} + \left(1 - \frac{\delta\eta_k}{s_k^T s_k}\right)\frac{s_k s_k^T}{y_k^T s_k + \delta\eta_k} \tag{17}$$

is again another rank two approximation to the inverse of the Hessian. Since the matrix $\bar{Q}_{k+1}$ defined by (17) is not symmetric and hence not positive definite, the corresponding directions are not necessarily descent and numerical instability can result. Observe that for $\delta = 0$, $\bar{Q}_{k+1} = Q_{k+1}$. With exact line searches ($s_k^T g_{k+1} = 0$), the direction d_{k+1} reduces to

$$d_{k+1} = -g_{k+1} + \frac{y_k^T g_{k+1}}{y_k^T s_k}\left(1 - \frac{\delta\eta_k}{y_k^T s_k + \delta\eta_k}\right) s_k$$

which is a modification of the Hestenes and Stiefel method. Besides, if $\delta = 0$, then we get exactly the Hestenes and Stiefel method.

The parameter θ_k given by (15) 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 θ_k selection, the direction d_{k+1} in (8) combines the HS and DY algorithms in a convex way. With these the following algorithm can be presented.

3. The HYBRIDM Algorithm

Step 1. Initialization. Select $x_0 \in R^n$, $\delta \ge 0$ 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. θ_k *parameter computation*. If $g_k^T g_{k+1} + \frac{g_k^T g_{k+1}}{y_k^T s_k}\delta\eta_k = 0$, then set $\theta_k = 0$, otherwise compute θ_k as in (15).

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

$$\left| g_{k+1}^T g_k \right| \geq 0.2 \left\| g_{k+1} \right\|^2 \tag{18}$$

is satisfied, then restart, i.e. set $d_{k+1} = -g_{k+1}$ otherwise define $d_{k+1} = d$. Compute the initial guess $\alpha_k = \alpha_{k-1} \left\| d_{k-1} \right\| / \left\| d_k \right\|$, set $k = k+1$ and continue with step 2. ■

Observe that for $\delta = 0$ we get the HYBRID algorithm [6]. 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 $-g_{k+1}$. Under reasonable assumptions, conditions (4), (5) and (18) 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 $k \geq 1$ the starting guess for the steplength α_k in the line search is computed as $\alpha_{k-1} \left\| d_{k-1} \right\|_2 / \left\| d_k \right\|_2$. This selection was used for the first time by Shanno and Phua in CONMIN [30]. It was also considered in the packages: SCG by Birgin and Martínez [9] and in SCALCG by Andrei [2,3,4].

4. Convergence Analysis

The global convergence properties of the nonlinear conjugate gradient methods with modified secant condition have been given by Yabe and Takano [34]. In the following we consider that $g_k \neq 0$ for all $k \geq 1$. Assume that:

(i) *The level set* $S = \left\{ x \in R^n : f(x) \leq f(x_0) \right\}$ *is bounded, i.e. there is a constant D such that* $\|x\| \leq D$ *for all* $x \in S$.

(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* $\left\| \nabla f(x) - \nabla f(y) \right\| \leq L \left\| x - y \right\|$*, for all* $x, y \in N$.

Under these assumptions on f there exists a constant $\Gamma \geq 0$ such that $\left\| \nabla f(x) \right\| \leq \Gamma$ for all $x \in S$. In order to prove the global convergence, we assume that the step size α_k in (2) is obtained by the strong Wolfe line search, that is,

$$f(x_k + \alpha_k d_k) - f(x_k) \leq \rho \alpha_k g_k^T d_k, \tag{19}$$

$$\left| g_{k+1}^T d_k \right| \leq \sigma g_k^T d_k. \tag{20}$$

where ρ and σ are positive constants such that $0 < \rho \leq \sigma < 1$.

Dai *et al.* [15] 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 α_k is obtained by the strong Wolfe line search (19) and (20). If*

$$\sum_{k\geq 1}\frac{1}{\|d_k\|^2}=\infty, \quad (21)$$

then

$$\liminf_{k\to\infty}\|g_k\|=0. \ \blacksquare \quad (22)$$

To prove the global convergence of the algorithm we need the following estimates. By the mean value theorem we have:

$$\begin{aligned}\eta_k &= 6(f_k-f_{k+1})+3(g_k+g_{k+1})^T s_k\\ &= 6\nabla f(\xi_k)^T(x_k-x_{k+1})+3(\nabla f(x_k)+\nabla f(x_{k+1}))^T s_k\\ &= -3\nabla f(\xi_k)^T s_k-3\nabla f(\xi_k)^T s_k+3\nabla f(x_k)^T s_k+3\nabla f(x_{k+1})^T s_k\\ &= 3\left(\nabla f(x_k)-\nabla f(\xi_k)+\nabla f(x_{k+1})-\nabla f(\xi_k)\right)^T s_k,\end{aligned}$$

where $\xi_k=\tau x_k+(1-\tau)x_{k+1}$ and $\tau\in(0,1)$. From the Lipschitz continuity we have:

$$\begin{aligned}|\eta_k| &\leq 3\left(\|\nabla f(x_k)-\nabla f(\xi_k)\|+\|\nabla f(x_{k+1})-\nabla f(\xi_k)\|\right)\|s_k\|\\ &\leq 3\left(L\|x_k-\xi_k\|+L\|x_{k+1}-\xi_k\|\right)\|s_k\|\\ &= 3\left(L(1-\tau)\|x_k-x_{k+1}\|+L\tau\|x_{k+1}-x_k\|\right)\|s_k\|\\ &= 3L(1-\tau)\|s_k\|^2+3L\tau\|s_k\|^2=3L\|s_k\|^2. \quad (23)\end{aligned}$$

On the other hand

$$\begin{aligned}\left|y_k^T s_k+\delta\eta_k\right| &\leq \left|y_k^T s_k\right|+\delta|\eta_k|\\ &\leq \|y_k\|\|s_k\|+\delta|\eta_k|\leq L\|s_k\|^2+3\delta L\|s_k\|^2=L(1+3\delta)\|s_k\|^2. \quad (24)\end{aligned}$$

Global convergence for uniformly convex functions. Suppose that $0<\theta_k<1$. For uniformly convex functions which satisfy the above assumptions *(i)* and *(ii)* we can prove that the norm of d_{k+1} generated by (8) and (15) is bounded above. Thus, by Lemma 1 we can prove the global convergence of the algorithm.

As we know, if f is a uniformly convex function, then there exists a constant $\mu>0$ such that

$$(\nabla f(x)-\nabla f(y))^T(x-y)\geq\mu\|x-y\|^2,\ \text{for any } x,y\in S. \quad (25)$$

Equivalently, this can be expressed as

$$f(x)\geq f(y)+\nabla f(y)^T(x-y)+\frac{\mu}{2}\|x-y\|^2,\ \text{for any } x,y\in S. \quad (26)$$

From (25) and (26) it follows that

$$y_k^T s_k\geq\mu\|s_k\|^2, \quad (27)$$

$$f_k-f_{k+1}\geq -g_{k+1}^T s_k+\frac{\mu}{2}\|s_k\|^2. \quad (28)$$

Obviously, from (27) and (28) we get:

$$\mu\|s_k\|^2\leq y_k^T s_k\leq L\|s_k\|^2, \quad (29)$$

i.e. $\mu\leq L.$

Theorem 1. *Suppose that the assumptions (i) and (ii) hold and* f *is a uniformly convex function. Consider the algorithm (2), (8) and (15), where* $0<\theta_k<1$, d_{k+1} *is a descent direction and* α_k *is obtained by the strong Wolfe line search (19) and (20). If* $L=\mu,$ *then for any* $\delta\geq 0$ *the algorithm satisfies* $\lim_{k\to\infty} g_k=0.$ *If* $L>\mu,$ *then for* $0\leq\delta\leq L/(3(L-\mu))$ *the algorithm satisfies* $\lim_{k\to\infty} g_k=0.$

Proof. Using the above relations (28) and (29) we have

$$\begin{aligned} y_k^T s_k+\delta\eta_k &= y_k^T s_k+6\delta(f_k-f_{k+1})+3\delta(g_k+g_{k+1})^T s_k \\ &\geq y_k^T s_k+6\delta(-g_{k+1}^T s_k+\frac{\mu}{2}\|s_k\|^2)+3\delta(g_k+g_{k+1})^T s_k \\ &= y_k^T s_k-6\delta g_{k+1}^T s_k+3\delta\mu\|s_k\|^2+3\delta g_k^T s_k+3\delta g_{k+1}^T s_k \\ &= (1-3\delta)y_k^T s_k+3\delta\mu\|s_k\|^2\geq(1-3\delta)y_k^T s_k+\frac{3\delta\mu}{L}y_k^T s_k \\ &= (1-3\delta+\frac{3\delta\mu}{L})y_k^T s_k. \end{aligned} \tag{30}$$

Now, if $L=\mu,$ then for all $\delta\geq 0$, $y_k^T s_k+\delta\eta_k\geq\mu\|s_k\|^2$, i.e. $y_k^T s_k+\delta\eta_k\geq m\|s_k\|^2$, where $m=\mu.$

On the other hand, if $L\geq\mu,$ then for $0\leq\delta<\frac{L}{3(L-\mu)}$, the coefficient of the right hand side of (30) is positive, that is $y_k^T s_k+\delta\eta_k\geq(1-3\delta+\frac{3\delta\mu}{L})\mu\|s_k\|^2$, i.e. $y_k^T s_k+\delta\eta_k\geq m\|s_k\|^2$, where $m=(1-3\delta+\frac{3\delta\mu}{L})\mu.$

Now, since $0<\theta_k<1$, using (15) in (8) after some algebra we have:

$$\begin{aligned} \|d_{k+1}\| &= \left\|-g_{k+1}+\frac{y_k^T g_{k+1}}{y_k^T s_k+\delta\eta_k}s_k-\left(1-\frac{\delta\eta_k}{\|s_k\|^2}\right)\frac{s_k^T g_{k+1}}{y_k^T s_k+\delta\eta_k}s_k\right\| \\ &\leq\|g_{k+1}\|+\frac{\|y_k\|\|g_{k+1}\|}{|y_k^T s_k+\delta\eta_k|}\|s_k\|+\left|1-\frac{\delta\eta_k}{\|s_k\|^2}\right|\frac{\|s_k\|\|g_{k+1}\|}{|y_k^T s_k+\delta\eta_k|}\|s_k\|. \end{aligned} \tag{31}$$

But, from (23) it follows that

$$\left|1-\frac{\delta\eta_k}{\|s_k\|^2}\right|\leq 1+\frac{\delta|\eta_k|}{\|s_k\|^2}\leq 1+\frac{\delta 3L\|s_k\|^2}{\|s_k\|^2}=1+3\delta L. \tag{32}$$

From (31), having in view the Lipschitz continuity, (32) and the above estimation on $y_k^T s_k+\delta\eta_k$ we get:

$$\begin{aligned} \|d_{k+1}\| &\leq\|g_{k+1}\|+\frac{L\|g_{k+1}\|}{m\|s_k\|^2}\|s_k\|^2+\left|1-\frac{\delta\eta_k}{\|s_k\|^2}\right|\frac{\|g_{k+1}\|}{m\|s_k\|^2}\|s_k\|^2 \\ &\leq\|g_{k+1}\|+\frac{L}{m}\|g_{k+1}\|+\frac{1+3\delta L}{m}\|g_{k+1}\| \end{aligned}$$

$$\le (1+\frac{L}{m}+\frac{1+3\delta L}{m})\Gamma. \tag{33}$$

This relation shows that

$$\sum_{k\ge 1}\frac{1}{\|d_k\|^2} \ge \left(\frac{m}{(m+L+1+3\delta L)\Gamma}\right)^2 \sum_{k\ge 1} 1 = \infty.$$

Therefore, from Lemma 1 we have $\liminf_{k\to\infty}\|g_k\| = 0,$ which for uniformly convex function is equivalent to $\lim_{k\to\infty} g_k = 0.$ ■

Observe that for $L > \mu$, $\frac{L}{3(L-\mu)} > \frac{1}{3}$. Theorem 1 says that there is a constant $\bar{\delta} > 1/3$ such that for any $0 \le \delta \le \bar{\delta}$, we have $\lim_{k\to\infty} g_k = 0.$

Global convergence for general nonlinear functions. Suppose that $0 < \theta_k < 1$. Using (15) in (7) from (6) we get the direction d_{k+1} as:

$$d_{k+1} = -g_{k+1} + \left[\frac{y_k^T g_{k+1}}{y_k^T s_k + \delta\eta_k} - \left(1 - \frac{\delta\eta_k}{\|s_k\|^2}\right)\frac{s_k^T g_{k+1}}{y_k^T s_k + \delta\eta_k}\right] s_k. \tag{34}$$

From (34) we see that if $0 < \theta_k < 1$, then

$$\beta_k^C = \frac{y_k^T g_{k+1}}{y_k^T s_k + \delta\eta_k} - \left(1 - \frac{\delta\eta_k}{\|s_k\|^2}\right)\frac{s_k^T g_{k+1}}{y_k^T s_k + \delta\eta_k}. \tag{35}$$

For general nonlinear functions, following the methods of Dai and Liao [12] or that of Yabe and Takano [34], we replace (35) by:

$$\beta_k^{C+} = \max\left\{\frac{y_k^T g_{k+1}}{y_k^T s_k + \delta\eta_k}, 0\right\} - \left(1 - \frac{\delta\eta_k}{\|s_k\|^2}\right)\frac{s_k^T g_{k+1}}{y_k^T s_k + \delta\eta_k} \tag{36}$$

and prove that the corresponding algorithm with strong Wolfe line search is globally convergent. Assume that the direction d_{k+1} satisfies the descent condition

$$g_{k+1}^T d_{k+1} \le 0. \tag{37}$$

To prove the global convergence by contradiction we assume that there is a positive constant γ such that

$$\|g_k\| \ge \gamma \text{ for all } k \ge 0. \tag{38}$$

Our analysis of (2), (6) and (36) for general nonlinear functions follows the insights developed by Gilbert and Nocedal in their analysis of the PRP+ conjugate gradient scheme [20] or that given by Hager and Zhang of their CG_DESCENT algorithm [21]. Similar to the approach considered by Yabe and Takano [34] we establish a bound for the change $w_{k+1} - w_k$ in the normalized direction $w_k = d_k / \|d_k\|$. This is used to conclude that the gradients cannot be bounded away from zero.

Lemma 2. *Suppose that the assumptions (i) and (ii) hold and consider the conjugate gradient algorithm (2), where* $0 < \theta_k < 1$, *the direction* d_{k+1} *given by (6) and (36) satisfies the descent* *condition (37) and* α_k *is obtained by the strong Wolfe line search conditions (19) and (20). If (38) holds and* δ *is chosen so that*

$$0 \le \delta < \frac{1-\sigma}{3(1+\sigma-2\rho)}$$

then $d_{k+1} \ne 0$ *and*

$$\sum_{k\ge 1} \left\| w_{k+1} - w_k \right\|^2 < \infty, \tag{39}$$

where $w_k = d_k / \left\| d_k \right\|$.

Proof. The proof is similar to that of Lemma 4 given in Yabe and Takano [34]. Obviously, by (37) we have $d_k \ne 0$. Therefore, w_k is well defined. Now, from (38) and Lemma 1 it follows that

$$\sum_{k\ge 0} \frac{1}{\left\| d_k \right\|^2} < \infty,$$

otherwise (22) holds, contradicting (38). In the following we write:

$$\beta_k^{C+} = \beta_k^{C1} + \beta_k^{C2}, \tag{40}$$

where:

$$\beta_k^{C1} = \max\left\{ \frac{y_k^T g_{k+1}}{y_k^T s_k + \delta \eta_k}, 0 \right\}, \tag{41}$$

$$\beta_k^{C2} = -\left(1 - \frac{\delta \eta_k}{\left\| s_k \right\|^2} \right) \frac{s_k^T g_{k+1}}{y_k^T s_k + \delta \eta_k}. \tag{42}$$

Define:

$$v_{k+1} = -g_{k+1} + \beta_k^{C2} s_k, \tag{43}$$

$$r_{k+1} = \frac{v_{k+1}}{\left\| d_{k+1} \right\|}, \tag{44}$$

$$\tau_{k+1} = \beta_k^{C1} \frac{\left\| d_k \right\|}{\left\| d_{k+1} \right\|} \ge 0. \tag{45}$$

Therefore, we have

$$\begin{aligned} w_{k+1} &= \frac{d_{k+1}}{\left\| d_{k+1} \right\|} = \frac{-g_{k+1} + \beta_k^{C1} s_k + \beta_k^{C2} s_k}{\left\| d_{k+1} \right\|} \\ &= \frac{-g_{k+1} + \beta_k^{C2} s_k}{\left\| d_{k+1} \right\|} + \beta_k^{C1} \frac{\left\| d_k \right\|}{\left\| d_{k+1} \right\|} \frac{s_k}{\left\| d_k \right\|} \\ &= r_{k+1} + \tau_{k+1} \alpha_k w_k. \end{aligned}$$

Now, since $\left\| w_k \right\| = \left\| w_{k+1} \right\| = 1,$ it follows that

$$\begin{aligned} \left\| r_{k+1} \right\|^2 &= \left\| w_{k+1} - \tau_{k+1} \alpha_k w_k \right\|^2 = \left\| w_{k+1} \right\|^2 - 2\tau_{k+1} \alpha_k w_{k+1}^T w_k + \tau_{k+1}^2 \alpha_k^2 \left\| w_k \right\|^2 \\ &= \left\| w_k \right\|^2 - 2\tau_{k+1} \alpha_k w_{k+1}^T w_k + \tau_{k+1}^2 \alpha_k^2 \left\| w_{k+1} \right\|^2 = \left\| \tau_{k+1} \alpha_k w_{k+1} - w_k \right\|^2. \end{aligned}$$

Therefore,

$$\left\| r_{k+1} \right\| = \left\| w_{k+1} - \tau_{k+1} \alpha_k w_k \right\| = \left\| \tau_{k+1} \alpha_k w_{k+1} - w_k \right\|.$$

Since $\tau_{k+1} \geq 0$ we get

$$\begin{aligned}\|w_{k+1}-w_k\| &\leq \|(1+\tau_{k+1}\alpha_k)(w_{k+1}-w_k)\| \\ &= \|w_{k+1}+\tau_{k+1}\alpha_k w_{k+1}-w_k-\tau_{k+1}\alpha_k w_k\| \\ &\leq \|w_{k+1}-\tau_{k+1}\alpha_k w_k\|+\|\tau_{k+1}\alpha_k w_{k+1}-w_k\| = 2\|r_{k+1}\|. \end{aligned} \tag{46}$$

Now, we evaluate the quantity $y_k^T s_k + \delta\eta_k$. Using the strong Wolfe conditions we have:

$$\begin{aligned} y_k^T s_k + \delta\eta_k &= y_k^T s_k + 6\delta(f_k - f_{k+1}) + 3\delta(g_k + g_{k+1})^T s_k \\ &\geq y_k^T s_k - 6\delta\rho g_k^T s_k + 3\delta(g_k+g_{k+1})^T s_k \\ &= (g_{k+1}-g_k)^T s_k - 6\delta\rho g_k^T s_k + 3\delta(g_k+g_{k+1})^T s_k \\ &= (1+3\delta) g_{k+1}^T s_k + (3\delta - 6\delta\rho - 1) g_k^T s_k \\ &\geq (1+3\delta)\sigma g_k^T s_k + (3\delta - 6\delta\rho - 1) g_k^T s_k \\ &= \left[3(1+\sigma-2\rho)\delta - (1-\sigma)\right] g_k^T s_k. \end{aligned} \tag{47}$$

We know that $g_k^T s_k = \alpha_k g_k^T d_k < 0$. Therefore, if $0 \leq \delta < \dfrac{1-\sigma}{3(1+\sigma-2\rho)}$, then there is a constant $M > 0$ such that

$$y_k^T s_k + \delta\eta_k \geq -M g_k^T s_k > 0. \tag{48}$$

From the definition of v_{k+1} it follows that

$$\begin{aligned}\|v_{k+1}\| &= \|-g_{k+1} + \beta_k^{C2} s_k\| \leq \|g_{k+1}\| + |\beta_k^{C2}| \|s_k\| \\ &= \|g_{k+1}\| + \left|1 - \frac{\delta\eta_k}{\|s_k\|^2}\right| \frac{|s_k^T g_{k+1}|}{|y_k^T s_k + \delta\eta_k|} \|s_k\| \\ &\leq \|g_{k+1}\| + \left|1 - \frac{\delta\eta_k}{\|s_k\|^2}\right| \frac{\sigma |s_k^T g_k|}{M |s_k^T g_k|} \|s_k\|. \end{aligned}$$

Therefore, using (32) we have

$$\|v_{k+1}\| \leq \|g_{k+1}\| + (1+3L\delta)\frac{\sigma}{M}\|s_k\| \leq \Gamma + (1+3L\delta)\frac{\sigma}{M} D. \tag{49}$$

With the above estimates we get:

$$\begin{aligned}\sum_{k\geq 1}\|w_{k+1}-w_k\|^2 &= \sum_{k\geq 1} 4\|r_k\|^2 = 4\sum_{k\geq 1}\frac{\|v_k\|^2}{\|d_k\|^2} \\ &\leq 4\left(\Gamma + (1+3L\delta)\frac{\sigma}{M}D\right)^2 \sum_{k\geq 1}\frac{1}{\|d_k\|^2} < \infty, \end{aligned}$$

i.e. (39) holds, which completes the proof. ■

This Lemma shows that asymptotically the search directions generated by the algorithm change slowly. Using Lemma 2 and assuming that d_k satisfies the sufficient descent condition

$$g_k^T d_k \leq -c\|g_k\|^2, \tag{50}$$

where $c > 0$ is a constant, we can establish the following lemma showing that β_k^{C+} satisfies a slightly different form of *Property (*)*. The Property (*), first derived by Gilbert and Nocedal

[20], shows that β_k in conjugate gradient algorithms will be small when the step s_k is small. For example, β_k^{PRP} has this property, this explaining the efficiency of the PRP conjugate gradient algorithm. Suppose that the step length α_k obtained by the strong Wolfe conditions (19) and (20) is bounded away from zero, i.e. there is a positive constant $\omega > 0$ such that $\alpha_k \geq \omega$.

Lemma 3. *Suppose that the assumptions (i) and (ii) hold and consider the conjugate gradient algorithm (2), where* $0 < \theta_k < 1$*, the direction* d_{k+1} *given by (6) and (36) satisfies the sufficient descent condition (50) and* α_k *is obtained by the strong Wolfe line search conditions (19) and (20) and* $\alpha_k \geq \omega$*. If* $0 \leq \delta < \dfrac{1-\sigma}{3(1+\sigma-2\rho)}$ *then there exist the constants* $b > 1$ *and* $\xi > 0$ *such that*

$$\left|\beta_k^{C+}\right| \leq b \tag{51}$$

and

$$\|s_k\| \leq \xi \Rightarrow \left|\beta_k^{C+}\right| \leq \frac{1}{b} \tag{52}$$

for all k.

Proof. From (48) and (38) we get:

$$y_k^T s_k + \delta\eta_k \geq -M g_k^T s_k \geq Mc\omega \|g_k\|^2 \geq Mc\omega\gamma^2. \tag{53}$$

Now, from (36), using (32) we have:

$$\begin{aligned}
\left|\beta_k^{C+}\right| &\leq \left|\frac{y_k^T g_{k+1}}{y_k^T s_k + \delta\eta_k}\right| + \left|1 - \frac{\delta\eta_k}{\|s_k\|^2}\right| \left|\frac{s_k^T g_{k+1}}{y_k^T s_k + \delta\eta_k}\right| \\
&\leq \frac{\left|y_k^T g_{k+1}\right| + (1+3\delta L)\left|s_k^T g_{k+1}\right|}{Mc\omega\gamma^2} \\
&\leq \frac{\|y_k\|\|g_{k+1}\| + (1+3\delta L)\|s_k\|\|g_{k+1}\|}{Mc\omega\gamma^2} \\
&\leq \frac{L+1+3\delta L}{Mc\omega\gamma^2}\|s_k\|\|g_{k+1}\| \leq \frac{L+1+3\delta L}{Mc\omega\gamma^2} D\Gamma \\
&= \frac{(L+1+3\delta L)D\Gamma}{Mc\omega\gamma^2} \equiv b.
\end{aligned} \tag{54}$$

Without loss of generality we can define b such that $b > 1$. Let us define:

$$\xi \equiv \left(\frac{Mc\omega\gamma^2}{(L+1+3\delta L)\Gamma}\right)^2 \frac{1}{D}. \tag{55}$$

Obviously, if $\|s_k\| \leq \xi$, from the fourth inequality in (54) we have

$$\left|\beta_k^{C+}\right| \leq \frac{(L+1+3\delta L)\Gamma}{Mc\omega\gamma^2}\xi = \frac{1}{b}.$$

Therefore, for b and ξ defined in (54) and (55) respectively, (51) and (52) hold. ■

The Property (*) presented in Lemma 3 can be used to show that if the gradients are bounded away from zero and (51) and (52) hold, then a finite number of steps s_k cannot be too small. Therefore, the algorithm makes a rapid progress to the optimum. Indeed, for $\lambda > 0$ and a positive integer Δ let us define the set of indices:

$$K_{k,\Delta}^{\lambda} = \left\{ i \in N^* : k \le i \le k + \Delta - 1, \|s_{i-1}\| > \lambda \right\},$$

where N^* is the set of positive integers. The following Lemma is similar to Lemma 3.5 in [12] and to Lemma 4.2 in [20].

Lemma 4. *Suppose that all the assumptions of Lemma 3 are satisfied. Then there is a* $\lambda > 0$ *such that for any* $\Delta \in N^*$ *and any index* k_0*, there is an index* $k \ge k_0$ *such that* $\left| K_{k,\Delta}^{\lambda} \right| > \Delta / 2.$

Using Lemma 2 and Lemma 4 we can prove the global convergence theorem for method (2), (6) and (36). The theorem is similar to Theorem 3.6 in Dai and Liao [12] or to Theorem 3.2 in Hager and Zhang [21] and the proof is omitted here.

Theorem 2. *Suppose that the assumptions (i) and (ii) hold and consider the conjugate gradient algorithm (2), where* $0 < \theta_k < 1$*, the direction* d_{k+1} *given by (6) and (36) satisfies the sufficient descent condition (50) and* α_k *is obtained by the strong Wolfe line search conditions (19) and (20). If* $0 \le \delta < \dfrac{1-\sigma}{3(1+\sigma-2\rho)}$ *then* $\liminf_{k\to\infty} \|g_k\| = 0.$ ■

Since ρ and σ are given in the Wolfe line search conditions, it follows that the upper bound of δ established in the Theorem 2 is smaller than 1/3. Although we were able to prove the global convergence of the hybrid computational scheme (2), (6) and (36), however, its computational performances are very close to those of the HYBRIDM variant. Therefore, in the next section we present only the results obtained with the HYBRIDM algorithm.

5. Numerical Experiments

In this section we present the computational performance of a Fortran implementation of the HYBRIDM algorithm on a set of 750 unconstrained optimization test problems. They are the unconstrained problems in the CUTE [10] 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 [17]. All algorithms implement the Wolfe line search conditions with $\rho = 0.0001$ and $\sigma = 0.9$. The same stopping criterion $\|g_k\|_\infty \le 10^{-6}$ is used, where $\|.\|_\infty$ is the maximum absolute component of a vector, and $\delta = 1.$ 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:

$$\left| f_i^{ALG1} - f_i^{ALG2} \right| < 10^{-3} \tag{56}$$

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 had the lowest functional value among the tested methods (up to 10^{-3} tolerance as it was specified in (56)). Clearly, this criterion is acceptable for users who are interested in minimizing functions and not in 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 HYBRIDM with the HYBRID conjugate gradient algorithm presented in [6]. Figure 1 presents the Dolan and Moré CPU performance profiles of HYBRIDM versus HYBRID. When comparing HYBRIDM with HYBRID (Figure 1) subject to the CPU time metric we see that HYBRIDM is top performer, i.e. the convex combination of HS and DY as expressed in (7) and (15) is more successful and more robust than the same convex combination using (10). Observe that out of 750 problems used in this numerical experiment only 727 satisfy (56). The percentage of the test problems for which a method is the fastest is given on the left axis of the plot. The right side of the plot gives the percentage of the test problems that were successfully solved by the HYBRID and HYBRIDM algorithms, respectively. Mainly, the right side is a measure of the robustness of an algorithm. Observe that the modified secant condition (13) is effective and gives a better approximation of $s_k^T \nabla^2 f(x_{k+1}) s_k$ by $s_k^T \hat{y}_k$ than the one given by $s_k^T y_k$.

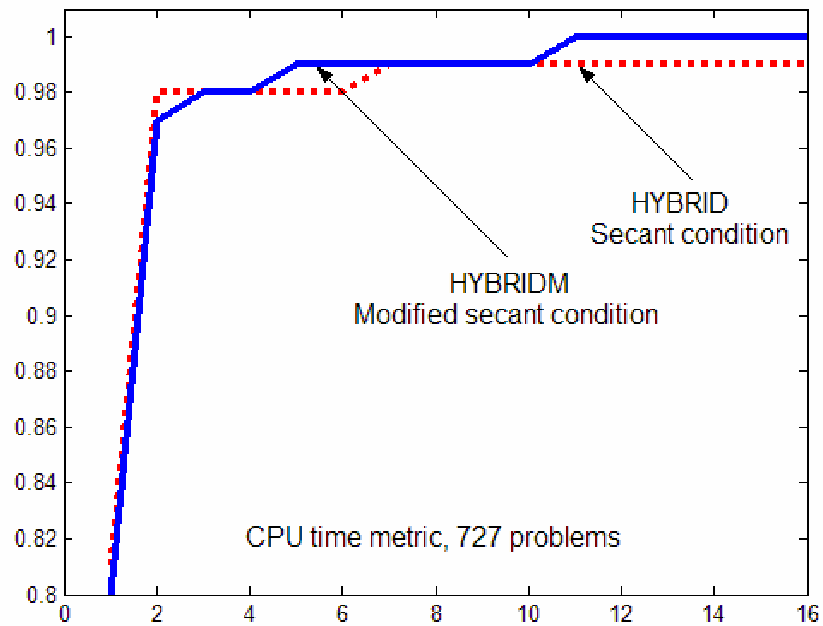

Figure 1. Performance based on CPU time. HYBRIDM versus HYBRID.

Table 2 presents the total number of iterations (#iter), the total number of function and gradient evaluations (#fg) and the total CPU time (seconds) for solving this set of 727 problems.

Table 2. Global performances

	HYBRID	**HYBRIDM**
#iter	242948	236020
#fg	1688489	1374097
CPU	767.41	726.02

Beside, we noticed that in contrast to the HYBRID algorithm which prefers to use the convex combination of HS and DY, HYBRIDM for the most of the iterations uses HS.

The second set of numerical experiments refers to the comparisons of HYBRIDM with the HS and the DY algorithms, respectively. Figures 2 and 3 present the Dolan and Moré CPU performance profiles of these algorithms.

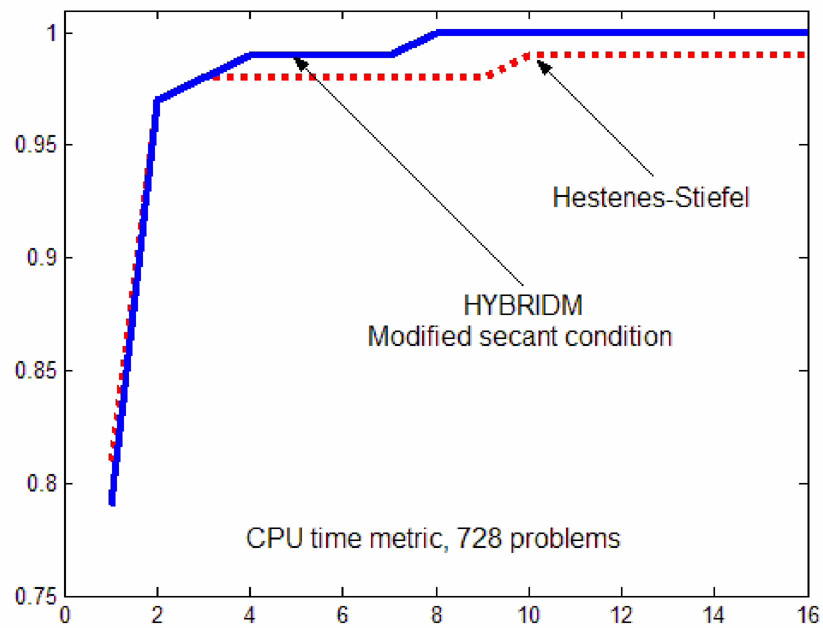

Figure 2. Performance based on CPU time. HYBRIDM versus Hestenes-Stiefel (HS).

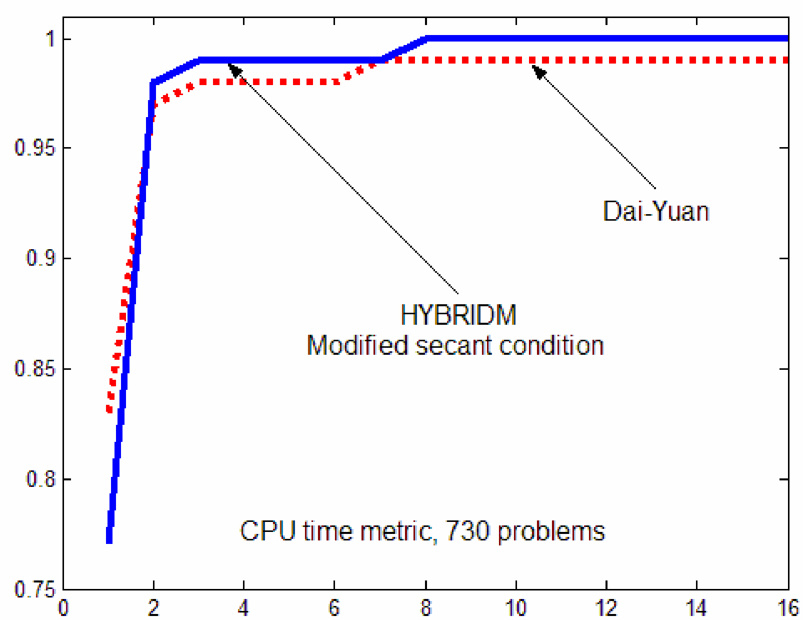

Figure 3. Performance based on CPU time. HYBRIDM versus Dai-Yuan (DY).

From the Figures above we see that HYBRIDM is again 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 the hybrid conjugate gradient algorithms HYBRIDM appears to generate the best search direction.

In the third set of numerical experiments we compare HYBRIDM with the hybrid variants of Dai and Yuan conjugate gradient algorithm hDY and hDYz, as in Figures 4 and 5 respectively.

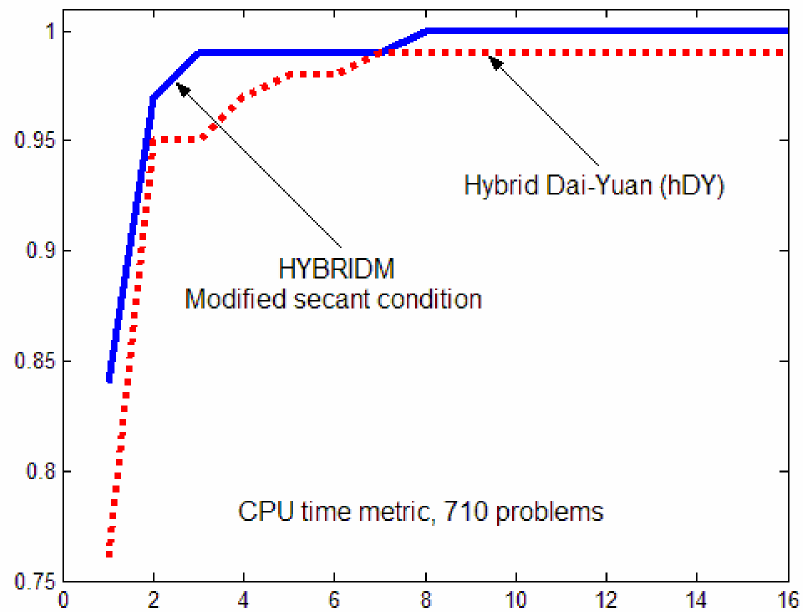

Figure 4. Performance based on CPU time. HYBRIDM versus hybrid Dai-Yuan (hDY).

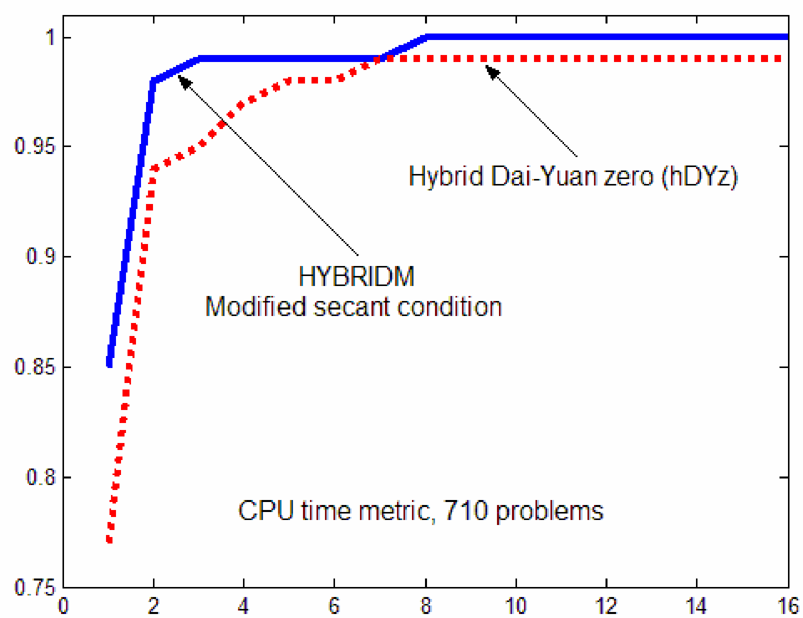

Figure 5. Performance based on CPU time. HYBRIDM versus hybrid Dai-Yuan zero (hDYz).

Observe that HYBRIDM is top performer among these conjugate gradient algorithms.

In the forth set of numerical experiments we compare HYBRIDM with SCALCG [2,3,4] as in Figure 6.

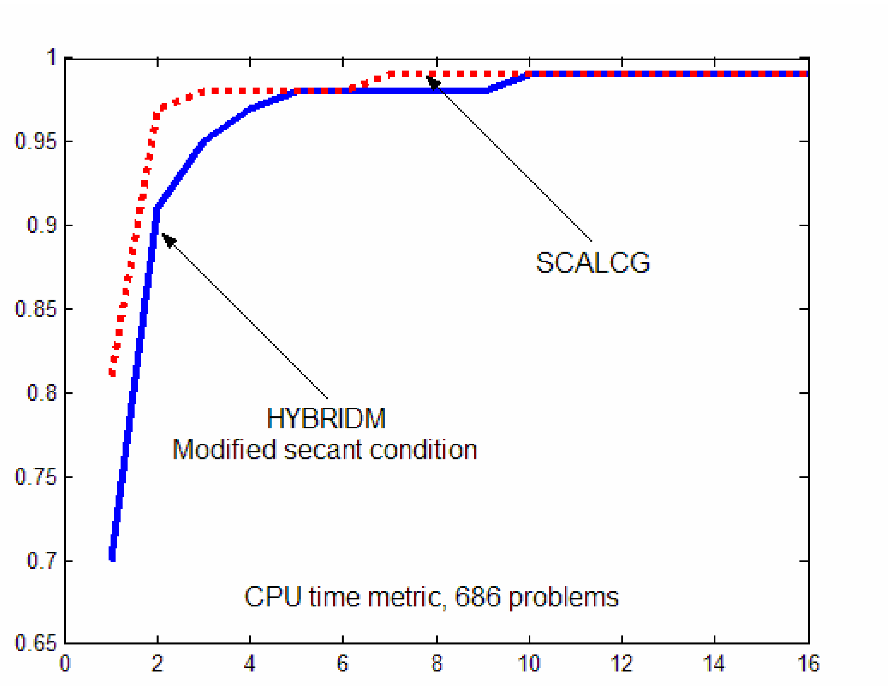

Figure 6. Performance based on CPU time. HYBRIDM versus SCALCG (θ spectral, Powell restart).

Observe that SCALCG using a BFGS preconditioned conjugate gradient algorithm is better than HYBRIDM.

In all our numerical experiments we have considered $\delta = 1$. However, the upper bound obtained in Theorem 1 for uniformly convex functions or that obtained in Theorem 2 for general nonlinear functions does not necessarily contain this value for δ. Therefore, further theoretical investigations must be done in order to get the optimal value for δ.

6. Conclusion

A large variety of conjugate gradient algorithms is well known. 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^{HS} and β_k^{DY}. The parameter in convex combination is computed in such a way so that the direction corresponding to this algorithm to be the Newton direction. Using the modified secant condition we get an algorithm which proved to be more efficient than the algorithm based on secant condition. For uniformly convex function our algorithm is globally convergent. For general nonlinear functions we proved the global convergence of a variant of the algorithm using the strong Wolfe line search.
The performance profile of our algorithm was higher than those of the well established conjugate gradient algorithms HS and DY and also of the hybrid variants hDY and hDYz, for a set of 750 unconstrained optimization problems. Additionally the proposed hybrid conjugate gradient algorithm is more robust than the HS and DY conjugate gradient algorithms.

REFERENCES

1. ANDREI, N., **An unconstrained optimization test functions collection**, Advanced Modeling and Optimization, 10 (2008), pp.147-161.

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

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

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

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

6. ANDREI, N., **Another hybrid conjugate gradient algorithm for unconstrained optimization,** Numerical Algorithms, 47 (2008), pp. 143-156.

7. ANDREI, N., **A hybrid conjugate gradient algorithm for unconstrained optimization. JOTA, Accepted.**

8. ANDREI, N., **New hybrid conjugate gradient algorithms as a convex combination of PRP and DY for unconstrained optimization,** ICI Technical Report, October 1, 2007.

9. BIRGIN, E. and J.M. MARTÍNEZ, **A spectral conjugate gradient method for unconstrained optimization,** Applied Math. and Optimization, 43, 2001, pp. 117-128.

10. BONGARTZ, I., A.R. CONN, N.I.M. GOULD, and P.L. TOINT, **CUTE: constrained and unconstrained testing environments**, ACM Trans. Math. Software, 21, 1995, pp. 123-160.

11. DAI, Y.H., **New properties of a nonlinear conjugate gradient method,** Numer. Math., 89 (2001), pp. 83-98.

12. DAI, Y.H. and L.Z. LIAO, **New conjugacy conditions and related nonlinear conjugate gradient methods,** Appl. Math. Optim., 43 (2001), pp. 87-101.

13. DAI, Y.H., and Y. YUAN, **A nonlinear conjugate gradient method with a strong global convergence property**, SIAM J. Optim., 10 (1999), pp. 177-182.

14. DAI, Y.H., and Y. YUAN, **An efficient hybrid conjugate gradient method for unconstrained optimization,** Ann. Oper. Res., 103 (2001), pp. 33-47.

15. DAI, Y.H., J.Y. HAN, G.H. LIU, D.F. SUN, X. YIN, and Y. YUAN, **Convergence properties of nonlinear conjugate gradient methods,** SIAM Journal on Optimization 10 (1999), pp. 348-358.

16. DANIEL, J.W., **The conjugate gradient method for linear and nonlinear operator equations,** SIAM J. Numer. Anal., 4 (1967), pp. 10-26.

17. DOLAN, E.D., and J.J. MORÉ, **Benchmarking optimization software with performance profiles,** Math. Programming, 91 (2002), pp. 201-213.

18. FLETCHER, R., **Practical Methods of Optimization, vol. 1: Unconstrained Optimization,** John Wiley & Sons, New York, 1987.

19. FLETCHER, R., and C. REEVES, **Function minimization by conjugate gradients**, Comput. J., 7 (1964), pp. 149-154.

20. GILBERT, J.C., and J. NOCEDAL, **Global convergence properties of conjugate gradient methods for optimization**, SIAM J. Optim., 2 (1992), pp. 21-42.

21. HAGER, W.W., and H. ZHANG, **A new conjugate gradient method with guaranteed descent and an efficient line search**, SIAM Journal on Optimization, 16(2005), pp. 170-192.

22. HAGER, W.W., and H. ZHANG, **A survey of nonlinear conjugate gradient methods**, Pacific journal of Optimization, 2 (2006), pp. 35-58.

23. HESTENES, M.R., and E.L. STIEFEL, **Methods of conjugate gradients for solving linear systems**, J. Research Nat. Bur. Standards, 49 (1952), pp. 409-436.

24. HU, Y.F. and C. STOREY, **Global convergence result for conjugate gradient methods**. J. Optim. Theory Appl., 71 (1991), pp. 399-405.

25. LIU, Y., and C. STOREY, **Efficient generalized conjugate gradient algorithms, Part 1: Theory**, JOTA, 69 (1991), pp. 129-137.

26. PERRY, A., **A modified conjugate gradient algorithm,** Discussion Paper no. 229, Center for Mathematical Studies in Economics and Management Science, Northwestern University, (1976).

27. POLAK, E., and G. RIBIERE, **Note sur la convergence de directions conjuguée,** Rev. Francaise Informat Recherche Operationelle, 3e Année 16 (1969), pp. 35-43.

28. POLYAK, B.T., **The conjugate gradient method in extreme problems**, USSR Comp. Math. Math. Phys., 9 (1969), pp. 94-112.

29. POWELL, M.J.D., **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.

30. SHANNO, D.F., and K.H. PHUA, **Algorithm 500, Minimization of unconstrained multivariate functions**, ACM Trans. on Math. Soft., 2, 1976, pp. 87-94.

31. TOUATI-AHMED, D., and C. STOREY, **Efficient hybrid conjugate gradient techniques**, J. Optim. Theory Appl., 64 (1990), pp. 379-397.

32. ZHANG, J.Z., 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.

33. ZHANG, J.Z., C.X. XU, **Properties and numerical performance of quasi-Newton methods with modified quasi-Newton equations,** Journal of Computational and Applied Mathematics, 137 (2001), pp. 269-278.

34. YABE, H., and M. TAKANO, **Global convergence properties of nonlinear conjugate gradient methods with modified secant condition,** Computational Optimization and Applications, 28 (2004), pp. 203-225.