Accelerated hybrid conjugate gradient algorithm with modified secant condition for unconstrained optimization

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

Abstract. An accelerated 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 [26]) and β_k^{DY} (Dai-Yuan [16]) 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 is the Newton direction and the pair (s_k, y_k) to satisfy the modified secant condition given by Li, Tang and Wei [28], $B_{k+1}s_k = z_k$, where $z_k = y_k + (\eta_k / \|s_k\|^2)s_k$, $\eta_k = 2(f_k - f_{k+1}) + (g_k + g_{k+1})^T s_k$, $s_k = x_{k+1} - x_k$ 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 [8], in which the pair (s_k, y_k) satisfies the classical secant condition $B_{k+1}s_k = y_k$, as well as some other conjugate gradient algorithms including Hestenes-Stiefel and Dai-Yuan. A set of 750 unconstrained optimization problems are used, some of them from the CUTE library [13].

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

1. Introduction

Let us consider the nonlinear unconstrained optimization problem

$$\min\left\{ f(x) : x \in R^n \right\}, \tag{1.1}$$

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

$$x_{k+1} = x_k + \alpha_k d_k, \tag{1.2}$$

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

$$d_{k+1} = -g_{k+1} + \beta_k d_k, \quad d_0 = -g_0. \tag{1.3}$$

In (1.3) 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:

$$f(x_k + \alpha_k d_k) - f(x_k) \le \rho \alpha_k g_k^T d_k, \tag{1.4}$$

$$\nabla f(x_k + \alpha_k d_k)^T d_k \ge \sigma g_k^T d_k, \tag{1.5}$$

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 (see [25]). The methods of Fletcher and Reeves (FR) [22], of Dai and Yuan (DY) [16] and the Conjugate Descent (CD) proposed by Fletcher [21]:

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

have strong convergence properties, but they may have modest practical performance due to jamming. On the other hand, the methods of Polak – Ribière [33] and Polyak (PRP) [34], of Hestenes and Stiefel (HS) [26] or of Liu and Storey (LS) [30]:

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

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 (1.2) and (1.3) 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 [17] (hDY)
2.	$\beta_k^{hDYz} = max\left\{0, min\left\{\beta_k^{HS}, \beta_k^{DY}\right\}\right\}$	Hybrid Dai-Yuan zero [17] (hDYz)
3.	$\beta_k^{GN} = max\left\{-\beta_k^{FR}, min\left\{\beta_k^{PRP}, \beta_k^{FR}\right\}\right\}$	Gilbert and Nocedal [23] (GN)
4.	$\beta_k^{HuS} = max\left\{0, min\left\{\beta_k^{PRP}, \beta_k^{FR}\right\}\right\}$	Hu and Storey [27] (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 [38] (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 [8] Convex combination of HS and DY with 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 [6, 9] Convex combination of PRP and DY with conjugacy condition
9.	$\beta_k^{AN} = (1-\theta_k)\beta_k^{PRP} + \theta_k\beta_k^{DY}$, $0 < \theta_k < 1$, $\theta_k = \frac{(y_k^T g_{k+1} - s_k^T g_{k+1})\lVert g_k \rVert^2 - (g_{k+1}^T y_k)(y_k^T s_k)}{\lVert g_{k+1} \rVert^2 \lVert g_k \rVert^2 - (g_{k+1}^T y_k)(y_k^T s_k)}$.	Andrei [6] Convex combination of PRP and DY with Newton direction

The hybrid computational schemes perform better than the classical conjugate gradient algorithms [5, 10]. In [8] we have presented a 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 [8]. In this paper, motivated by a result given by Li, Tang and Wei [28] 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 classical secant condition.

The structure of the paper is as follows. Section 2 introduces our hybrid conjugate gradient algorithm, AHYBRIDM as a convex combination of HS and DY algorithms with modified secant condition. Section 3 presents the convergence of this hybrid conjugate gradient computational scheme and in section 4 the algorithm and its acceleration is shown. In section 5 some numerical experiments and performance profiles of Dolan-Moré [20] 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 [13] as well as some other ones presented in [7]. It is shown that this hybrid conjugate gradient algorithm outperforms the classical HS and DY conjugate gradient algorithms and also the some other conjugate gradient algorithms including hybrid variants hDY, hDYz, GN and LS-CD.

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 (1.2), where the stepsize $\alpha_k > 0$ is determined according to the Wolfe line search conditions (1.4) and (1.5), and the directions d_k are generated by the rule:

$$d_{k+1} = -g_{k+1} + \beta_k^C s_k, \; d_0 = -g_0, \tag{2.1}$$

where

$$\beta_k^C = (1-\theta_k)\beta_k^{HS} + \theta_k \beta_k^{DY} = (1-\theta_k)\frac{g_{k+1}^T y_k}{y_k^T s_k} + \theta_k \frac{g_{k+1}^T g_{k+1}}{y_k^T s_k} \tag{2.2}$$

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 [36], 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, 10].

On the other hand, the DY method always generates descent directions, and in [14] 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 (2.1) and (2.2) it is obvious that

$$d_{k+1} = -g_{k+1} + (1-\theta_k)\frac{y_k^T g_{k+1}}{y_k^T s_k} s_k + \theta_k \frac{g_{k+1}^T g_{k+1}}{y_k^T s_k} s_k . \tag{2.3}$$

Our motivation is to choose the parameter θ_k in such a way so that the direction d_{k+1} given by (2.3) to be the Newton direction. 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:

$$\theta_k = \frac{s_k^T \nabla^2 f(x_{k+1}) g_{k+1} - s_k^T g_{k+1} - \dfrac{y_k^T g_{k+1}}{y_k^T s_k} s_k^T \nabla^2 f(x_{k+1}) s_k}{\left[\dfrac{g_{k+1}^T g_{k+1}}{y_k^T s_k} - \dfrac{y_k^T g_{k+1}}{y_k^T s_k}\right] s_k^T \nabla^2 f(x_{k+1}) s_k}. \tag{2.4}$$

However, in this formula the salient point is the presence of the Hessian. One of the first conjugate gradient algorithm using the Hessian was given by Daniel [19] where $\beta_k = (g_{k+1}^T \nabla^2 f(x_k) d_k)/(d_k^T \nabla^2 f(x_k) d_k)$. For large-scale problems, choices for the update parameter that do not require the evaluation of the Hessian matrix are often preferred in practice to the methods that require the Hessian.

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 [8] 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 (see [8]), where:

$$\theta_k = -\frac{s_k^T g_{k+1}}{g_k^T g_{k+1}} . \tag{2.5}$$

Zhang, Deng and Chen [39] 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 (2.3) and (2.5), where $0 < \theta_k < 1$, is an approximation of the Newton direction. Observe that if $0 < \theta_k < 1$, then our direction can be expressed as:

$$d_{k+1} = -Q_{k+1} g_{k+1}, \tag{2.6}$$

where

$$Q_{k+1} = I - \frac{s_k y_k^T}{y_k^T s_k} + \frac{s_k s_k^T}{y_k^T s_k} \tag{2.7}$$

is a rank two approximation to the inverse of the Hessian. It is worth saying that the matrix Q_{k+1} was first proposed by Perry [32]. 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 (2.7) 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

(2.4) to get (2.5). 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. In [8] we have computational evidence that our HYBRID algorithm is top performer versus HS, DY, hDY and hDYz conjugate gradient algorithms.

Li, Tang and Wei [28] expanded the secant condition and obtained a modified secant condition which uses both the gradients and the function values in two successive points as:

$$B_{k+1} s_k = z_k, \quad z_k = y_k + \frac{\eta_k}{\|s_k\|^2} s_k, \tag{2.8}$$

where $\eta_k = 2(f_k - f_{k+1}) + (g_k + g_{k+1})^T s_k$. Obviously, from (2.8) we get

$$s_k^T B_{k+1} s_k = s_k^T y_k + \eta_k. \tag{2.9}$$

Theorem 2.1. *If $f(x)$ is a smooth general nonlinear function, then when $\|s_k\| \to 0$,*

$$s_k^T \nabla^2 f(x_{k+1}) s_k - s_k^T y_k = O(\|s_k\|^3),$$
$$s_k^T \nabla^2 f(x_{k+1}) s_k - s_k^T z_k = O(\|s_k\|^4).$$

The proof is similar to that given in Theorem 1.1 by Zhang, Deng and Chen [39] and is omitted here. ■

Therefore, the quantity $s_k^T z_k$ given by the modified secant condition (2.8) 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 (2.4). For this purpose, in order to unify both approaches, we consider a slight modification of the modified secant condition (2.8) as $B_{k+1} s_k = z_k$, where

$$z_k = y_k + \frac{\delta \eta_k}{\|s_k\|^2} s_k$$

and $\delta \geq 0$ is a scalar parameter. This leads us to another hybrid conjugate gradient algorithm (1.2), (2.1) and (2.2), where

$$\theta_k = \frac{\left(\frac{\delta \eta_k}{s_k^T s_k} - 1\right) s_k^T g_{k+1} - \frac{y_k^T g_{k+1}}{y_k^T s_k} \delta \eta_k}{g_k^T g_{k+1} + \frac{g_k^T g_{k+1}}{y_k^T s_k} \delta \eta_k}. \tag{2.10}$$

Therefore, motivated by the theorem 2.1, the direction (2.3) and (2.10), where $0 < \theta_k < 1$, is a better approximation of the Newton direction than that given by using (2.5) in (2.4). Now, using (2.10) in (2.3) we get

$$d_{k+1} = -g_{k+1} + \frac{y_k^T g_{k+1}}{y_k^T s_k + \delta \eta_k} s_k - \left(1 - \frac{\delta \eta_k}{\|s_k\|^2}\right) \frac{s_k^T g_{k+1}}{y_k^T s_k + \delta \eta_k} s_k \tag{2.11}$$

Theorem 2.2. *Assume that f is a convex function and α_k in algorithm (1.2) and (2.3), where θ_k is given by (2.10), is determined by the Wolfe line search (1.4) and (1.5). If $0 < \theta_k < 1$ and $\delta \leq \|s_k\|^2 / \eta_k$ then the direction d_{k+1} given by (2.11) is a descent one.*

Proof. From (2.11) we get

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

The second term in (2.12) can be written as

$$\frac{(y_k^T g_{k+1})(s_k^T g_{k+1})}{y_k^T s_k + \delta\eta_k} = \frac{(y_k^T g_{k+1})(y_k^T s_k + \delta\eta_k)(s_k^T g_{k+1})}{(y_k^T s_k + \delta\eta_k)^2} =$$

$$\frac{[(y_k^T s_k + \delta\eta_k)g_{k+1}]^T[(s_k^T g_{k+1})y_k]}{(y_k^T s_k + \delta\eta_k)^2} \le \frac{(y_k^T s_k + \delta\eta_k)^2\|g_{k+1}\|^2 + (s_k^T g_{k+1})^2\|y_k\|^2}{2(y_k^T s_k + \delta\eta_k)^2} =$$

$$\frac{1}{2}\|g_{k+1}\|^2 + \frac{(s_k^T g_{k+1})^2\|y_k\|^2}{2(y_k^T s_k + \delta\eta_k)^2}. \tag{2.13}$$

Now, using (2.13) in (2.12) we get

$$g_{k+1}^T d_{k+1} \le -\left(\frac{1}{2}\|g_{k+1}\|^2 + \left(1 - \frac{\delta\eta_k}{\|s_k\|^2}\right)\frac{(s_k^T g_{k+1})^2}{y_k^T s_k + \delta\eta_k}\right) + \frac{(s_k^T g_{k+1})^2\|y_k\|^2}{2(y_k^T s_k + \delta\eta_k)^2}. \tag{2.14}$$

From Wolfe line search and (2.9) observe that $y_k^T s_k + \delta\eta_k \ge 0$. Besides, $1 - \delta\eta_k / \|s_k\|^2 \ge 0$. Observe that the last term in (2.14) tends to zero very fast. Therefore, $g_{k+1}^T d_{k+1} \le 0$, i.e. d_{k+1} is a descent direction. ■

Remark 1. Since the last term in (2.14) tends to zero very fast, it can be neglected. Besides, observe that $\left(1 - \frac{\delta\eta_k}{\|s_k\|^2}\right)\frac{(s_k^T g_{k+1})^2}{y_k^T s_k + \delta\eta_k}$ also tends to zero very fast. Therefore, the direction d_{k+1} satisfies the sufficient descent condition $g_{k+1}^T d_{k+1} \le -c\|g_{k+1}\|$, where c is a positive constant, and $c \approx 1/2$.

As above, observe that if $0 < \theta_k < 1$, then our direction can be expressed as:

$$d_{k+1} = -\bar{Q}_{k+1} g_{k+1}, \tag{2.15}$$

where

$$\bar{Q}_{k+1} = I - \frac{s_k y_k^T}{y_k^T s_k + \delta\eta_k} + \left(1 - \frac{\delta\eta_k}{s_k^T s_k}\right)\frac{s_k s_k^T}{y_k^T s_k + \delta\eta_k} \tag{2.16}$$

is again another rank two approximation to the inverse of the Hessian. Since the matrix $\bar{Q}_{k+1}$ defined by (2.16) is not symmetric and hence not positive definite, again the corresponding directions are not necessarily descent and numerical instability can result. However, this is more elaborated than Q_{k+1} in (2.7). 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

$$d_{k+1} = -g_{k+1} + \frac{y_k^T g_{k+1}}{y_k^T s_k + \delta\eta_k} s_k,$$

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 (2.10) can be outside the interval $[0,1]$. However, in order to have a real convex combination in (2.2) the following rule is considered: if $\theta_k \le 0$, then set $\theta_k = 0$ in (2.2), i.e. $\beta_k^C = \beta_k^{HS}$; on the other hand if $\theta_k \ge 1$, then take $\theta_k = 1$ in

(2.2), i.e. $\beta_k^C = \beta_k^{DY}$. Therefore, under this rule for θ_k selection, the direction d_{k+1} in (2.3) combines the HS and DY algorithms in a convex way.

3. Convergence analysis

In the following we consider that $g_k \neq 0$ for all $k \geq 1$. Assume that:

(i) *The level set* $S = \{x \in R^n : f(x) \leq f(x_0)\}$ *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* $\|\nabla f(x) - \nabla f(y)\| \leq L\|x - y\|$, *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 order to prove the global convergence, we assume that the step size α_k in (1.2) is obtained by the strong Wolfe line search, that is,

$$f(x_k + \alpha_k d_k) - f(x_k) \leq \rho \alpha_k g_k^T d_k, \tag{3.1}$$

$$\left|\nabla f(x_{k+1})^T d_k\right| \leq \sigma g_k^T d_k. \tag{3.2}$$

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

Dai *et al.* [18] proved that for any conjugate gradient method with strong Wolfe line search the following general result holds:

Lemma 3.1. *Suppose that the assumptions (i) and (ii) hold and consider any conjugate gradient method (1.2) and (1.3), where* d_k *is a descent direction and* α_k *is obtained by the strong Wolfe line search (3.1) and (3.2). If*

$$\sum_{k \geq 1} \frac{1}{\|d_k\|^2} = \infty, \tag{3.3}$$

then

$$\liminf_{k \to \infty} \|g_k\| = 0. \ \blacksquare \tag{3.4}$$

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

$$\begin{aligned}
\eta_k &= 2(f_k - f_{k+1}) + (g_k + g_{k+1})^T s_k \\
&= 2\nabla f(\xi_k)^T (x_k - x_{k+1}) + (\nabla f(x_k) + \nabla f(x_{k+1}))^T s_k \\
&= -\nabla f(\xi_k)^T s_k - \nabla f(\xi_k)^T s_k + \nabla f(x_k)^T s_k + \nabla f(x_{k+1})^T s_k \\
&= \left(\nabla f(x_k) - \nabla f(\xi_k) + \nabla f(x_{k+1}) - \nabla f(\xi_k)\right)^T s_k,
\end{aligned}$$

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

$$\begin{aligned}
|\eta_k| &\leq \left(\|\nabla f(x_k) - \nabla f(\xi_k)\| + \|\nabla f(x_{k+1}) - \nabla f(\xi_k)\|\right)\|s_k\| \\
&\leq \left(L\|x_k - \xi_k\| + L\|x_{k+1} - \xi_k\|\right)\|s_k\| \\
&= \left(L(1-\tau)\|x_k - x_{k+1}\| + L\tau\|x_{k+1} - x_k\|\right)\|s_k\| \\
&= L(1-\tau)\|s_k\|^2 + L\tau\|s_k\|^2 = L\|s_k\|^2.
\end{aligned} \tag{3.5}$$

On the other hand

$$\left|y_k^T s_k + \delta \eta_k\right| \leq \left|y_k^T s_k\right| + \delta |\eta_k|$$

$$\le \|y_k\|\|s_k\| + \delta|\eta_k| \le L\|s_k\|^2 + \delta L\|s_k\|^2 = L(1+\delta)\|s_k\|^2. \tag{3.6}$$

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 (2.3) and (2.10) is bounded above. Thus, by Lemma 3.1 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

$$(\nabla f(x) - \nabla f(y))^T (x-y) \ge \mu\|x-y\|^2, \text{ for any } x, y \in S. \tag{3.7}$$

Equivalently, this can be expressed as

$$f(x) \ge f(y) + \nabla f(y)^T (x-y) + \frac{\mu}{2}\|x-y\|^2, \text{ for any } x, y \in S. \tag{3.8}$$

From (3.7) and (3.8) it follows that

$$y_k^T s_k \ge \mu\|s_k\|^2, \tag{3.9}$$

$$f_k - f_{k+1} \ge -g_{k+1}^T s_k + \frac{\mu}{2}\|s_k\|^2. \tag{3.10}$$

Obviously, from (3.9) and (3.10) we get:

$$\mu\|s_k\|^2 \le y_k^T s_k \le L\|s_k\|^2, \tag{3.11}$$

i.e. $\mu \le L$.

Theorem 3.1. *Suppose that the assumptions (i) and (ii) hold and f is a uniformly convex function. Consider the algorithm (1.2), (2.3) and (2.10), where $0 < \theta_k < 1$, d_{k+1} is a descent direction and α_k is obtained by the strong Wolfe line search (3.1) and (3.2). If $L = \mu$, then for any $\delta \ge 0$ the algorithm satisfies $\lim_{k\to\infty} g_k = 0$. If $L > \mu$, then for $0 \le \delta \le L/(L-\mu)$ the algorithm satisfies $\lim_{k\to\infty} g_k = 0$.*

Proof. Using the above relations (3.10) and (3.11) we have

$$\begin{aligned} y_k^T s_k + \delta\eta_k &= y_k^T s_k + 2\delta(f_k - f_{k+1}) + \delta(g_k + g_{k+1})^T s_k \\ &\ge y_k^T s_k + 2\delta(-g_{k+1}^T s_k + \frac{\mu}{2}\|s_k\|^2) + \delta(g_k + g_{k+1})^T s_k \\ &= y_k^T s_k - 2\delta g_{k+1}^T s_k + \delta\mu\|s_k\|^2 + \delta g_k^T s_k + \delta g_{k+1}^T s_k \\ &= (1-\delta) y_k^T s_k + \delta\mu\|s_k\|^2 \ge (1-\delta) y_k^T s_k + \frac{\delta\mu}{L} y_k^T s_k \\ &= (1-\delta+\frac{\delta\mu}{L}) y_k^T s_k. \end{aligned} \tag{3.12}$$

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

On the other hand, if $L \ge \mu$, then for $0 \le \delta < \frac{L}{L-\mu}$, the coefficient of the right hand side of (3.12) is positive, that is $y_k^T s_k + \delta\eta_k \ge (1-\delta+\frac{\delta\mu}{L})\mu\|s_k\|^2$, i.e. $y_k^T s_k + \delta\eta_k \ge m\|s_k\|^2$, where $m = (1-\delta+\frac{\delta\mu}{L})\mu$.

Now, from (2.11) we have:

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

$$\leq \|g_{k+1}\| + \frac{\|y_k\|\|g_{k+1}\|}{|y_k^T s_k + \delta\eta_k|}\|s_k\| + \left|1 - \frac{\delta\eta_k}{\|s_k\|^2}\right| \frac{\|s_k\|\|g_{k+1}\|}{|y_k^T s_k + \delta\eta_k|}\|s_k\|. \tag{3.13}$$

But, from (3.5) it follows that

$$\left|1 - \frac{\delta\eta_k}{\|s_k\|^2}\right| \leq 1 + \frac{\delta|\eta_k|}{\|s_k\|^2} \leq 1 + \frac{\delta L\|s_k\|^2}{\|s_k\|^2} = 1 + \delta L. \tag{3.14}$$

From (3.13), having in view the Lipschitz continuity, (3.14) and the above estimation on $y_k^T s_k + \delta\eta_k$ we get:

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

$$\leq \|g_{k+1}\| + \frac{L}{m}\|g_{k+1}\| + \frac{1+\delta L}{m}\|g_{k+1}\|$$

$$\leq (1 + \frac{L}{m} + \frac{1+\delta L}{m})\Gamma. \tag{3.15}$$

This relation shows that

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

Therefore, from Lemma 3.1 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 $L > \mu$, $\frac{L}{L-\mu} > 1$. Theorem 3.1 says that there is a constant $\bar{\delta} > 1$ such that for any $\delta \leq \bar{\delta}$, we have $\lim_{k\to\infty} g_k = 0.$

Global convergence for general nonlinear functions. From (2.11) we see that if $0 < \theta_k < 1$, then

$$\beta_k^C = \frac{y_k^T g_{k+1}}{y_k^T s_k + \delta\eta_k} - \left(1 - \frac{\delta\eta_k}{\|s_k\|^2}\right) \frac{s_k^T g_{k+1}}{y_k^T s_k + \delta\eta_k}. \tag{3.16}$$

For general nonlinear functions, we replace (3.16) by:

$$\beta_k^{C+} = \max\left\{\frac{y_k^T g_{k+1}}{y_k^T s_k + \delta\eta_k}, 0\right\} - \left(1 - \frac{\delta\eta_k}{\|s_k\|^2}\right) \frac{s_k^T g_{k+1}}{y_k^T s_k + \delta\eta_k} \tag{3.17}$$

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 (see Theorem 2.2)

$$g_{k+1}^T d_{k+1} \leq 0. \tag{3.18}$$

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

$$\|g_k\| \geq \gamma \text{ for all } k \geq 0. \tag{3.19}$$

Our analysis of (1.2), (2.1) and (3.17) for general nonlinear functions follows the insights developed by Gilbert and Nocedal in their analysis of the PRP+ conjugate gradient scheme [23] or that given by Hager and Zhang of their CG_DESCENT algorithm [24].

Lemma 3.2. *Suppose that the assumptions (i) and (ii) hold and consider the conjugate gradient algorithm (1.2), where* $0<\theta_k<1$, *the direction* d_{k+1} *given by (2.1) and (3.17) satisfies the descent condition (3.18) and* α_k *is obtained by the strong Wolfe line search conditions (3.1) and (3.2). If (3.19) holds and* δ *is chosen so that*

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

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

$$\sum_{k\ge 1} \left\| w_{k+1} - w_k \right\|^2 < \infty, \tag{3.20}$$

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

Proof. Obviously, by (3.18) we have $d_k \neq 0$. Therefore, w_k is well defined. Now, from (3.19) and Lemma 3.1 it follows that

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

otherwise (3.4) holds, contradicting (3.19). In the following we write:

$$\beta_k^{C+} = \beta_k^{C1} + \beta_k^{C2}, \tag{3.21}$$

where:

$$\beta_k^{C1} = \max\left\{ \frac{y_k^T g_{k+1}}{y_k^T s_k + \delta\eta_k}, 0 \right\}, \tag{3.22}$$

$$\beta_k^{C2} = -\left(1 - \frac{\delta\eta_k}{\left\| s_k \right\|^2} \right) \frac{s_k^T g_{k+1}}{y_k^T s_k + \delta\eta_k}. \tag{3.23}$$

Define:

$$v_{k+1} = -g_{k+1} + \beta_k^{C2} s_k, \tag{3.24}$$

$$r_{k+1} = \frac{v_{k+1}}{\left\| d_{k+1} \right\|}, \tag{3.25}$$

$$\tau_{k+1} = \beta_k^{C1} \frac{\left\| d_k \right\|}{\left\| d_{k+1} \right\|} \ge 0. \tag{3.26}$$

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

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

$$= \|w_k\|^2 - 2\tau_{k+1}\alpha_k w_{k+1}^T w_k + \tau_{k+1}^2 \alpha_k^2 \|w_{k+1}\|^2 = \|\tau_{k+1}\alpha_k w_{k+1} - w_k\|^2 .$$

Therefore,

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

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

$$\begin{aligned}
\|w_{k+1} - w_k\| &\leq \|(1+\tau_{k+1}\alpha_k)(w_{k+1} - w_k)\| \\
&= \|w_{k+1} + \tau_{k+1}\alpha_k w_{k+1} - w_k - \tau_{k+1}\alpha_k w_k\| \\
&\leq \|w_{k+1} - \tau_{k+1}\alpha_k w_k\| + \|\tau_{k+1}\alpha_k w_{k+1} - w_k\| = 2\|r_{k+1}\|. \qquad (3.27)
\end{aligned}$$

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

$$\begin{aligned}
y_k^T s_k + \delta\eta_k &= y_k^T s_k + 2\delta(f_k - f_{k+1}) + \delta(g_k + g_{k+1})^T s_k \\
&\geq y_k^T s_k - 2\delta\rho g_k^T s_k + \delta(g_k + g_{k+1})^T s_k \\
&= (g_{k+1} - g_k)^T s_k - 2\delta\rho g_k^T s_k + \delta(g_k + g_{k+1})^T s_k \\
&= (1+\delta) g_{k+1}^T s_k + (\delta - 2\delta\rho - 1) g_k^T s_k \\
&\geq (1+\delta)\sigma g_k^T s_k + (\delta - 2\delta\rho - 1) g_k^T s_k \\
&= \left[(1+\sigma-2\rho)\delta - (1-\sigma)\right] g_k^T s_k . \qquad (3.28)
\end{aligned}$$

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

$$y_k^T s_k + \delta\eta_k \geq -M g_k^T s_k > 0. \qquad (3.29)$$

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 (3.14) we have

$$\|v_{k+1}\| \leq \|g_{k+1}\| + (1+L\delta)\frac{\sigma}{M}\|s_k\| \leq \Gamma + (1+L\delta)\frac{\sigma}{M} D. \qquad (3.30)$$

With the above estimates we get:

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

i.e. (3.20) holds, which completes the proof. ■

This Lemma shows that asymptotically the search directions generated by the algorithm change slowly. Using Lemma 3.2 and assuming that d_k satisfies the sufficient descent condition (see remark 1)

$$g_k^T d_k \leq -c\|g_k\|^2 , \qquad (3.31)$$

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 [23], 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 (3.1) and (3.2) is bounded away from zero, i.e. there is a positive constant $\omega > 0$ such that $\alpha_k \geq \omega$. Dai and Liao [15] proved that this property is responsible for the global convergence of conjugate gradient algorithms.

Lemma 3.3. *Suppose that the assumptions (i) and (ii) hold and consider the conjugate gradient algorithm (1.2), where* $0 < \theta_k < 1$, *the direction* d_{k+1} *given by (2.1) and (3.17) satisfies the sufficient descent condition (3.31) and* α_k *is obtained by the strong Wolfe line search conditions (3.1) and (3.2) with* $\alpha_k \geq \omega$. *If* $0 \leq \delta < \dfrac{1-\sigma}{(1+\sigma-2\rho)}$ *then there exist the constants* $b > 1$ *and* $\xi > 0$ *such that*

$$\left|\beta_k^{C+}\right| \leq b \tag{3.32}$$

and

$$\|s_k\| \leq \xi \Rightarrow \left|\beta_k^{C+}\right| \leq \frac{1}{b} \tag{3.33}$$

for all k.

Proof. From (3.29), (3.31) and (3.19) we get:

$$y_k^T s_k + \delta\eta_k \geq -Mg_k^T s_k \geq Mc\omega\|g_k\|^2 \geq Mc\omega\gamma^2. \tag{3.34}$$

Now, from (3.17), using (3.14) we have:

$$\begin{aligned}
\left|\beta_k^{C+}\right| &\leq \left|\frac{y_k^T g_{k+1}}{y_k^T s_k + \delta\eta_k}\right| + \left|1 - \frac{\delta\eta_k}{\|s_k\|^2}\right| \left|\frac{s_k^T g_{k+1}}{y_k^T s_k + \delta\eta_k}\right| \\
&\leq \frac{\left|y_k^T g_{k+1}\right| + (1+\delta L)\left|s_k^T g_{k+1}\right|}{Mc\omega\gamma^2} \\
&\leq \frac{\|y_k\|\|g_{k+1}\| + (1+\delta L)\|s_k\|\|g_{k+1}\|}{Mc\omega\gamma^2} \\
&\leq \frac{L+1+\delta L}{Mc\omega\gamma^2}\|s_k\|\|g_{k+1}\| \leq \frac{L+1+\delta L}{Mc\omega\gamma^2} D\Gamma \equiv b.
\end{aligned} \tag{3.35}$$

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

$$\xi \equiv \left(\frac{Mc\omega\gamma^2}{(L+1+\delta L)\Gamma}\right)^2 \frac{1}{D}. \tag{3.36}$$

Obviously, if $\|s_k\| \leq \xi$, from the fourth inequality in (3.35) we have

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

Therefore, for b and ξ defined in (3.35) and (3.36) respectively, (3.32) and (3.33) hold. ■

The Property (*) presented in Lemma 3.3 can be used to show that if the gradients are bounded away from zero and (3.32) and (3.33) 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 $\tau > 0$ and a positive integer Δ let us define the set of indices:

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

where N^* is the set of positive integers. The following Lemma is similar to Lemma 3.5 in [15] and to Lemma 4.2 in [23].

Lemma 3.4. *Suppose that all the assumptions of Lemma 3.3 are satisfied. Then there is a* $\tau > 0$ *such that for any* $\Delta \in N^*$ *and any index* k_0*, there is an index* $k \ge k_0$ *such that* $\left| K_{k,\Delta}^{\tau} \right| > \Delta / 2.$

Using Lemma 3.2 and Lemma 3.4 we can prove the global convergence theorem for method (1.2), (2.1) and (3.17). The theorem is similar to Theorem 3.6 in Dai and Liao [15] or to Theorem 3.2 in Hager and Zhang [24] and the proof is omitted here.

Theorem 3.2. *Suppose that the assumptions (i) and (ii) hold and consider the conjugate gradient algorithm (1.2), where* $0 < \theta_k < 1$*, the direction* d_{k+1} *given by (2.1) and (3.17) satisfies the sufficient descent condition (3.31) and* α_k *is obtained by the strong Wolfe line search conditions (3.1) and (3.2). If* $0 \le \delta < \dfrac{1-\sigma}{(1+\sigma-2\rho)}$ *then* $\liminf_{k\to\infty} \left\| g_k \right\| = 0.$ ■

Since ρ and σ are given in the Wolfe line search conditions, it follows that the upper bound of δ established in the Theorem 3.2 is smaller than 1.

4. The AHYBRIDM algorithm

In [31] Nocedal pointed out that in conjugate gradient methods the step lengths may differ from 1 in a very unpredictable manner. They can be larger or smaller than 1 depending on how the problem is scaled. This is in very sharp contrast to the Newton and quasi-Newton methods, including the limited memory quasi-Newton methods, which accept the unit steplength most of the time along the iterations, and therefore usually they require only few function evaluations per search direction. Numerical comparisons between conjugate gradient methods and the limited memory quasi Newton method by Liu and Nocedal [29], show that the latter is more successful [5, 10]. One explanation of efficiency of this limited memory quasi-Newton method is given by its ability to accept unity step lengths along the iterations. In this section we take advantage of this behavior of conjugate gradient algorithms and consider an acceleration scheme we have presented in [1, 11]. Basically the acceleration scheme modifies the step length α_k in a multiplicative manner to improve the reduction of the function values along the iterations (see [1] and [11]). In accelerated algorithm instead of (1.2) the new estimation of the minimum point is computed as

$$x_{k+1} = x_k + \lambda_k \alpha_k d_k, \tag{4.1}$$

where

$$\lambda_k = -\frac{a_k}{b_k}, \tag{4.2}$$

$a_k = \alpha_k g_k^T d_k$, $b_k = -\alpha_k (g_k - g_z)^T d_k$, $g_z = \nabla f(z)$ and $z = x_k + \alpha_k d_k$. Hence, if $b_k \ne 0$, then the new estimation of the solution is computed as $x_{k+1} = x_k + \lambda_k \alpha_k d_k$, otherwise $x_{k+1} = x_k + \alpha_k d_k$. Therefore, using the definitions of g_k, s_k, y_k and the above acceleration scheme (4.1) and (4.2) we can present the following hybrid conjugate gradient algorithm.

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

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

Step 3. Line search. Compute $\alpha_k > 0$ satisfying the Wolfe line search conditions (1.4) and (1.5).

Step 4. Compute: $z = x_k + \alpha_k d_k$, $g_z = \nabla f(z)$ and $y_k = g_k - g_z$.

Step 5. Compute: $a_k = \alpha_k g_k^T d_k$, and $b_k = -\alpha_k y_k^T d_k$.

Step 6. Acceleration scheme. If $b_k \neq 0$, then compute $\lambda_k = -a_k / b_k$ and update the variables as $x_{k+1} = x_k + \lambda_k \alpha_k d_k$, otherwise update the variables as $x_{k+1} = x_k + \alpha_k d_k$. Compute f_{k+1} and g_{k+1}. Compute $s_k = x_{k+1} - x_k$, $y_k = g_{k+1} - g_k$ and $\eta_k = 2(f_k - f_{k+1}) + (g_k + g_{k+1})^T s_k$.

Step 7. θ_k parameter computation. If $g_k^T g_{k+1} + \dfrac{g_k^T g_{k+1}}{y_k^T s_k} \delta \eta_k = 0$, then set $\theta_k = 0$, otherwise compute θ_k as in (2.10).

Step 8. β_k^C conjugate gradient parameter computation. If $0 < \theta_k < 1$, then compute β_k^C as in (2.2). 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 9. Direction computation. Compute $d = -g_{k+1} + \beta_k^C s_k$. If the restart criterion of Powell

$$\left| g_{k+1}^T g_k \right| \geq 0.2 \|g_{k+1}\|^2 \tag{4.3}$$

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 α_k satisfying the Wolfe line search conditions (1.4) and (1.5). 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 [35], but we are interested in the performance of a conjugate gradient algorithm that uses this restart criterion. Under reasonable assumptions, conditions (1.4), (1.5) and (4.3) 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 $\alpha_{k-1} \|d_{k-1}\|_2 / \|d_k\|_2$. This selection was used for the first time by Shanno and Phua in CONMIN [37]. It was also considered in the packages: SCG by Birgin and Martínez [12] and in SCALCG by Andrei [2,3,4].

5. Numerical experiments

In this section we report the computational performance of a Fortran implementation of the AHYBRIDM algorithm on a set of 750 unconstrained optimization test problems. 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: $n = 1000, 2000, \ldots, 10000$ (see [7]). Comparisons with other conjugate gradient algorithms, including the performance profiles of Dolan and Moré [20] are presented. 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, 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:

$$\left| f_i^{ALG1} - f_i^{ALG2} \right| < 10^{-3} \tag{5.1}$$

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 (5.1)). 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 AHYBRIDM with the HYBRID conjugate gradient algorithm presented in [8]. Figure 1 shows the Dolan and Moré CPU performance profiles of AHYBRIDM versus HYBRID.

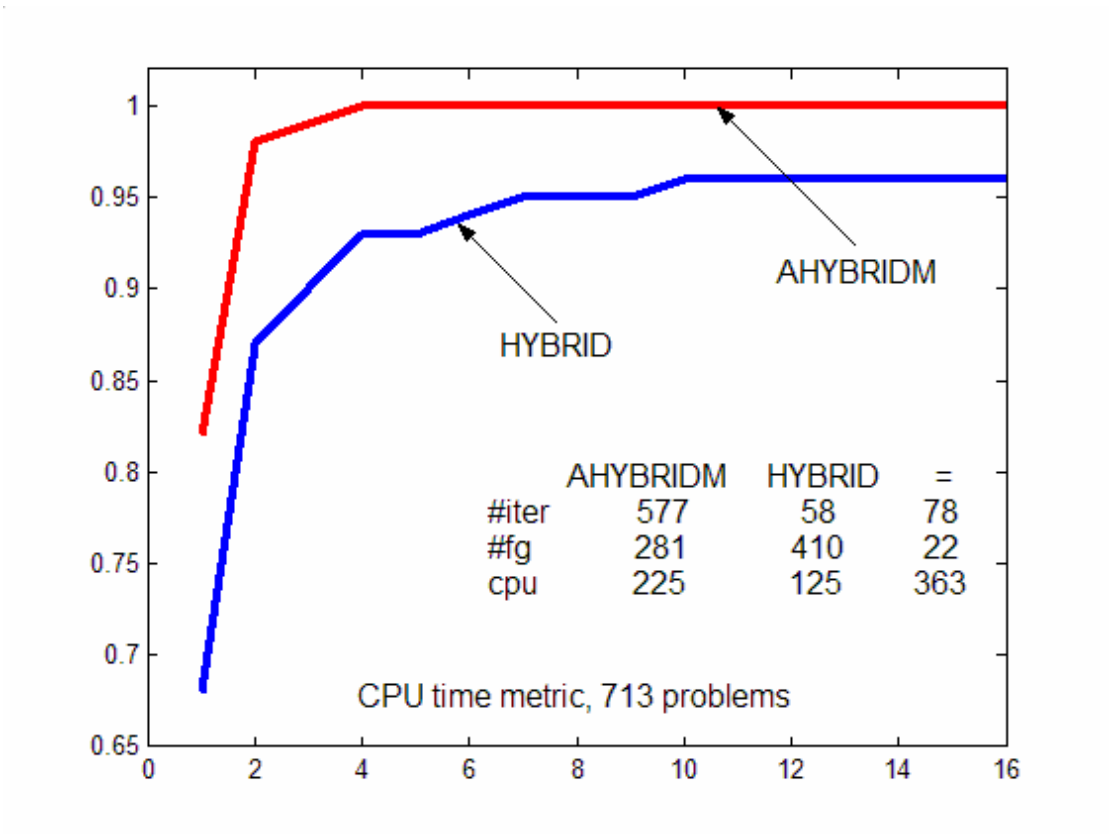

Fig. 1. Performance based on CPU time. AHYBRIDM versus HYBRID [8].

When comparing AHYBRIDM with HYBRID (Figure 1) subject to the CPU time metric we see that AHYBRIDM is top performer, i.e. the convex combination of HS and DY as expressed in (2.2) and (2.10) is more successful and more robust than the same convex combination using (2.5). We see that subject to the number of iterations, AHYBRIDM was better in 577 problems (i.e. it achieved the minimum number of iterations in 577 problems), HYBRID was better in 58 problems and they achieved the same number of iterations in 78 problems, etc. Observe that out of 750 problems used in this numerical experiment only 713 satisfy (5.1). 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 AHYBRIDM algorithms, respectively. Mainly, the right side is a measure of the robustness of an algorithm. Observe that the modified secant condition (2.8) is effective and gives a better approximation of $s_k^T \nabla^2 f(x_{k+1}) s_k$ by $s_k^T z_k$ than the one given by $s_k^T y_k$. Besides, the acceleration scheme used in AHYBRIDM algorithm has a major role. It is worth saying that in unconstrained optimization all the efforts concentrate on the search direction computation. In our approach, besides this, we try to improve the algorithms by modifying the steplength α_k (computed by the Wolfe line search conditions) through an acceleration scheme.

The second set of numerical experiments refers to the comparisons of AHYBRIDM with the HS and the DY algorithms, respectively. Figure 2 presents the Dolan and Moré CPU time performance profiles of these algorithms.

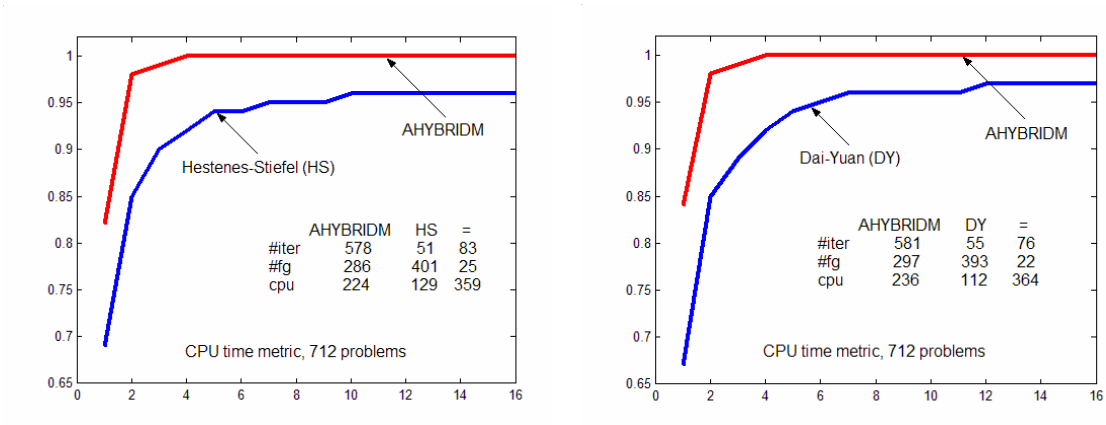

Fig. 2. Performance based on CPU time. AHYBRIDM versus HS and DY.

From the plots in Figure 2 we see that AHYBRIDM is again top performer. We see that this convex combination of HS and DY algorithms combined with the acceleration scheme lead us to a more efficient conjugate gradient algorithm. Both the modified secant condition (2.8) and the acceleration scheme (4.1)-(4.2) implemented in AHYBRIDM are important ingredients in getting an efficient conjugate gradient algorithm.

In the third set of numerical experiments we compare AHYBRIDM with PRP (Polak-Ribière-Polyak) and LS (Liu and Storey) classical conjugate gradient algorithms. Figure 3 presents the Dolan and Moré performance profiles of these algorithms.

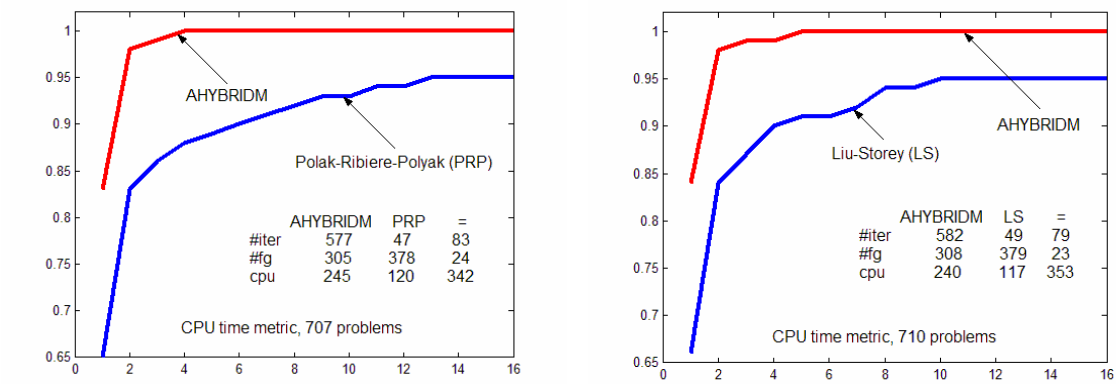

Fig. 3. Performance based on CPU time. HYBRIDM versus PRP and LS.

In the fourth set of numerical experiments we compare AHYBRIDM with the hybrid conjugate gradient algorithms hDY, hDYz, GN and LS-CD (see Table 1) as in Figures 4.

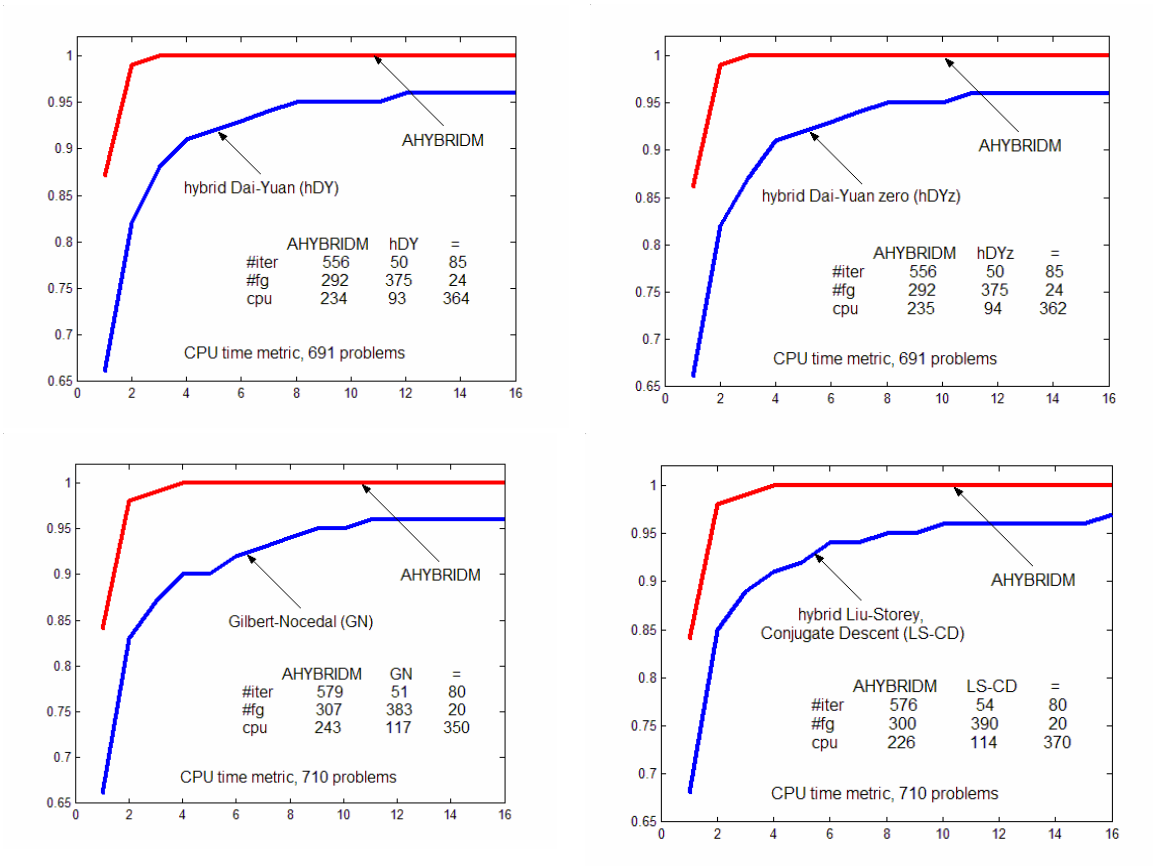

Fig. 4. Performance based on CPU time. HYBRIDM versus hDY, hDYz, GN and LS-CD.

Observe that AHYBRIDM is top performer among the conjugate gradient algorithms and the differences are substantial.

In all our numerical experiments we have considered $\delta = 1$. However, the upper bound obtained in Theorem 3.1 for uniformly convex functions or that obtained in Theorem 3.2 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. For $\delta = 0$ we get an accelerated variant of HYBRID algorithm presented in [8].

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 generates descent direction and proved to be more efficient than the algorithm based on the classical 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 PRP and LS and of the known hybrid variants hDY, hDYz, GN and LS-CD 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 acceleration of gradient descent algorithm with backtracking for unconstrained optimization. Numerical Algorithms, 42 (2006), pp.63-73.

[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), 561-571.

[4] Andrei, N., A scaled BFGS preconditioned conjugate gradient algorithm for unconstrained optimization. Applied Mathematics Letters, 20 (2007), 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., New hybrid conjugate gradient algorithms as a convex combination of PRP and DY for unconstrained optimization. ICI Technical Report, October 1, 2007.

[7] Andrei, N., An unconstrained optimization test functions collection. Advanced Modeling and Optimization, (2008) 10, pp.147-161.

[8] Andrei, N., Another hybrid conjugate gradient algorithm for unconstrained optimization. Numerical Algorithms. 47 (2008), 143-156.

[9] Andrei, N., A hybrid conjugate gradient algorithm for unconstrained optimization. JOTA, *Accepted.*

[10] Andrei, N., *Performance profiles of conjugate gradient algorithms for unconstrained optimization.* Encyclopedia of Optimization, 2nd edition, C.A. Floudas and P.M. Pardalos (Eds.), Springer, New York, vol. P (2009), 2938-2953.

[11] Andrei, N., *Accelerated conjugate gradient algorithm with finite difference Hessian / vector product approximation for unconstrained optimization.* Journal of Computational and Applied Mathematics, *Accepted.*

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

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

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

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

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

[17] Dai, Y.H. Yuan, Y., An efficient hybrid conjugate gradient method for unconstrained optimization, Ann. Oper. Res., 103 (2001), pp. 33-47.

[18] Dai, Y.H. 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.

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

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

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

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

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

[24] Hager, W.W., Zhang, H., A new conjugate gradient method with guaranteed descent and an efficient line search, SIAM Journal on Optimization, 16 (2005) 170-192.

[25] Hager, W.W., Zhang, H., A survey of nonlinear conjugate gradient methods. Pacific journal of Optimization, 2 (2006), pp.35-58.

[26] Hestenes, M.R., Stiefel, E.L., *Methods of conjugate gradients for solving linear systems*, J. Research Nat. Bur. Standards, 49 (1952), pp.409-436.
[27] Hu, Y.F., Storey, C., *Global convergence result for conjugate gradient methods.* J. Optim. Theory Appl., 71 (1991), pp.399-405.
[28] Li, G., Tang, C., Wei, Z., New conjugacy condition and related new conjugate gradient methods for unconstrained optimization. Journal of Computational and Applied Mathematics, 202 (2007), 523-539.
[29] Liu, D.C. and Nocedal, J., *On the limited memory BFGS method for large scale optimization.* Mathematical Programming, 45 (1989), pp.503-528.
[30] Liu, Y., Storey, C., *Efficient generalized conjugate gradient algorithms, Part 1: Theory.* JOTA, 69 (1991), pp.129-137.
[31] Nocedal, J., *Conjugate gradient methods and nonlinear optimization,* in L. Adams and J.L. Nazareth (Eds.) *Linear and Nonlinear Conjugate Gradient – Related Methods,* SIAM, Philadelphia, 1996, pp.9-23.
[32] Perry, A., *A modified conjugate gradient algorithm.* Discussion Paper no. 229, Center for Mathematical Studies in Economics and Management Science, Northwestern University, (1976).
[33] Polak, E., Ribière, G., *Note sur la convergence de directions conjuguée,* Rev. Francaise Informat Recherche Operationelle, 3e Année 16 (1969), pp.35-43.
[34] Polyak, B.T., *The conjugate gradient method in extreme problems*. USSR Comp. Math. Math. Phys., 9 (1969), pp.94-112.
[35] Powell, M.J.D., *Restart procedures for the conjugate gradient method.* Mathematical Programming 12 (1977), pp.241-254.
[36] 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.
[37] Shanno, D.F., Phua, K.H., *Algorithm 500, Minimization of unconstrained multivariate functions*, ACM Trans. on Math. Soft., 2, pp.87-94, 1976.
[38] Touati-Ahmed, D., Storey, C., *Efficient hybrid conjugate gradient techniques.* J. Optim. Theory Appl., 64 (1990), pp.379-397.
[39] Zhang, J.Z., Deng, N.Y., Chen, L.H., *New quasi-Newton equation and related methods for unconstrained optimization.* J. Optim. Theory Appl., 102 (1999), pp.147-167.

February 23, 2009