ANOTHER CONJUGATE GRADIENT ALGORITHM FOR UNCONSTRAINED OPTIMIZATION

Another hybrid conjugate gradient algorithm is proposed and analyzed. The parameter β_k is computed as a convex combination of β_k^{HS} corresponding to Hestenes-Stiefel and β_k^{DY} of Dai-Yuan conjugate gradient algorithms, i.e. $\beta_k^C = (1-\theta_k)\beta_k^{HS} + \theta_k \beta_k^{DY}$. The parameter θ_k is computed in such a way that the direction corresponding to the conjugate gradient algorithm is equating the Newton direction. The algorithm uses the standard Wolfe line search conditions. Numerical comparisons with conjugate gradient algorithms using a set of 750 unconstrained optimization problems, some of them from the CUTE library, show that this hybrid computational scheme outperforms the Hestenes-Stiefel and the Dai-Yuan conjugate gradient algorithms, as well as some other known conjugate gradient algorithms.

Introduction. For solving the nonlinear unconstrained optimization problem

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

where $f : R^n \to R$ is a continuously differentiable function, bounded from below, starting from an initial guess $x_0 \in R^n$, a nonlinear conjugate gradient method, generates a sequence $\{x_k\}$ as:

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

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

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

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

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

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

where d_k is a descent direction and $0 < \rho \le \sigma < 1$. Plenty of conjugate gradient methods are known, and an excellent survey of these methods, with a special attention on their global convergence, is given by Hager and Zhang [17]. Different conjugate gradient algorithms correspond to different choices for the scalar parameter β_k. Methods Fletcher and Reeves (FR) [14], Dai and Yuan (DY) [11] and Conjugate Descent (CD) proposed by Fletcher [13]:

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

have strong convergence properties, but they may have modest practical performance due to jamming. On the other hand, the methods of Polak – Ribière [20] and Polyak (PRP) [21], Hestenes and Stiefel (HS) [18] or Liu and Storey (LS) [19]:

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

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

In order to exploit the attractive features of each set, the so called hybrid conjugate gradient methods have been proposed. The known hybrid conjugate gradient methods, summarized in

[5], combine in a *projective* manner the above conjugate gradient methods. In this paper we suggest another approach based on a *convex combination* of conjugate gradient algorithms.

The hybrid conjugate gradient algorithm as a convex combination of HS and DY algorithms. Our algorithm generates the iterates $x_0, x_1, x_2, \ldots$ computed by means of the recurrence (2), where the stepsize $\alpha_k > 0$ is determined according to the Wolfe conditions (4) and (5), and the directions d_k are generated by the rule:

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

where

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

and θ_k is a scalar parameter satisfying $0 \le \theta_k \le 1$, which follows to be determined. Observe that if $\theta_k = 0$, then $\beta_k^C = \beta_k^{HS}$, 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 [22], the HS method with the exact line search, for general nonlinear functions, may not converge. The HS method has a built-in restart feature that addresses directly to the jamming phenomenon. Indeed, when the step $x_{k+1} - x_k$ is small, then the factor $y_k = g_{k+1} - g_k$ in the numerator of 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 [17].

The DY method, on the other side, always generates descent directions, and in [8] Dai established a remarkable property for the DY conjugate gradient algorithm, relating the descent directions to the sufficient descent condition. It is shown that if there exist constants 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 $\le j$.

From (6) and (7) it is easy to see that

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

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

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

having in view that $\nabla^2 f(x_{k+1}) s_k = y_k$, after some algebra, we get

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

Theorem 1. *Assume that d_k is a descent direction and α_k in algorithm (2) and (8) where θ_k is given by (9) is determined by the Wolfe line search (4) and (5). If $0 < \theta_k < 1$, and*

$$\frac{(y_k^T g_{k+1})(s_k^T g_{k+1})}{y_k^T s_k} \le \|g_{k+1}\|^2, \tag{10}$$

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

Proof. From (8) and (9) we get

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

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

$$\frac{s_k^T g_{k+1}}{y_k^T s_k} < 1. \tag{12}$$

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

$$0 < 1 + \frac{s_k^T g_{k+1}}{g_k^T g_{k+1}} < 1. \tag{13}$$

Therefore, from (10) we have

$$\begin{aligned} g_{k+1}^T d_{k+1} &\le -\left[1 + \frac{(s_k^T g_{k+1})^2}{(g_k^T g_{k+1})(y_k^T s_k)}\right]\|g_{k+1}\|^2 + \left[1 + \frac{s_k^T g_{k+1}}{g_k^T g_{k+1}}\right]\|g_{k+1}\|^2 \\ &= -\left(\frac{s_k^T g_{k+1}}{g_k^T g_{k+1}}\right)\left[\frac{s_k^T g_{k+1}}{y_k^T s_k} - 1\right]\|g_{k+1}\|^2 \le 0. \end{aligned} \tag{14}$$

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

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

$$g_{k+1}^T d_{k+1} \le -\delta \|g_{k+1}\|^2, \tag{15}$$

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

Proof. From (14) we have

$$g_{k+1}^T d_{k+1} \le -\left(\frac{s_k^T g_{k+1}}{g_k^T g_{k+1}}\right)\left(\frac{s_k^T g_k}{y_k^T s_k}\right)\|g_{k+1}\|^2. \tag{16}$$

Since $y_k^T s_k > 0$ and $s_k^T g_k \le 0$, it follows that there exists a constant $c_2 > 0$, such that $g_k^T s_k \le -c_2 (y_k^T s_k) < 0$. On the other hand, since $1 > \theta_k \ge c_1 > 0$, then $s_k^T g_{k+1} \le -c_1 (g_k^T g_{k+1})$. Therefore, from (16) we have

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

where $\delta = c_1 c_2 > 0$. ■

The parameter θ_k given by (9) can be outside the interval $[0,1]$. However, in order to have a real convex combination in (7) the following rule is 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 in a convex manner the HS and DY algorithms.

The NDHSDY algorithm

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

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

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

Step 4. θ_k parameter computation. If $g_k^T g_{k+1} = 0$, then set $\theta_k = 0$, otherwise compute θ_k as in (9).

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

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

$$\left| g_{k+1}^T g_k \right| \ge 0.2 \|g_{k+1}\|^2, \tag{17}$$

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 (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 (17) are sufficient to prove the global convergence of the algorithm.

The first trial of the step length crucially affects the practical behavior of the algorithm. At every iteration $k \ge 1$ the starting guess for the step α_k in the line search is computed as $\alpha_{k-1}\|d_{k-1}\|_2 / \|d_k\|_2$. This selection was considered for the first time by Shanno and Phua in CONMIN [23]. It is also considered in the packages: SCG by Birgin and Martínez [6] and in SCALCG by Andrei [2,3,4].

Convergence analysis. Assume that:

(i) *The level set $S = \{x \in R^n : f(x) \le f(x_0)\}$ is bounded.*

(ii) *In a neighborhood N of S, the function f is continuously differentiable and its gradient is Lipschitz continuous, i.e. there exists a constant $L > 0$ such that $\|\nabla f(x) - \nabla f(y)\| \le L\|x - y\|$, for all $x, y \in N$.*

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

In [10] it is proved that for any conjugate gradient method with strong Wolfe line search the following general result holds:

Lemma 1. *Suppose that the assumptions (i) and (ii) hold and consider any conjugate gradient method (2) and (3), where d_k is a descent direction and α_k is obtained by the strong Wolfe line search*

$$f(x_k + \alpha_k d_k) - f(x_k) \le \rho \alpha_k g_k^T d_k, \tag{18}$$

$$\left| g_{k+1}^T d_k \right| \le \sigma g_k^T d_k. \tag{19}$$

If

$$\sum_{k\geq 1}\frac{1}{\|d_k\|^2}=\infty, \qquad (20)$$

then

$$\liminf_{k\to\infty}\|g_k\|=0. \ \blacksquare \qquad (21)$$

For uniformly convex functions which satisfy the above assumptions we can prove that the norm of d_{k+1} generated by (8) and (9) is bounded above. Thus, by Lemma 1 we have the following result.

Theorem 3. *Suppose that the assumptions (i) and (ii) hold. Consider the algorithm (2) and (8)-(9), where d_{k+1} is a descent direction and α_k is obtained by the strong Wolfe line search (18) and (19). If for $k\geq 0$, $0<\theta_k<1$ and there exists the nonnegative constant η_1 such that*

$$\|g_{k+1}\|^2\leq \eta_1\|s_k\|, \qquad (22)$$

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

$$(\nabla f(x)-\nabla f(y))^T(x-y)\geq \mu\|x-y\|^2, \qquad (23)$$

then

$$\lim_{k\to\infty} g_k=0. \qquad (24)$$

Proof. From (23) it follows that $y_k^T s_k\geq \mu\|s_k\|^2$. Now, since $0<\theta_k<1$, from uniform convexity and (22) we have:

$$\left|\beta_k^C\right|\leq\left|\frac{y_k^T g_{k+1}}{y_k^T s_k}\right|+\left|\frac{g_{k+1}^T g_{k+1}}{y_k^T s_k}\right|\leq\frac{\|g_{k+1}\|\|y_k\|}{\mu\|s_k\|^2}+\frac{\eta_1\|s_k\|}{\mu\|s_k\|^2}. \qquad (25)$$

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

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

Hence, with (25) we have

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

which implies that (20) is true. Therefore, by Lemma 1 we have (21), which for uniformly convex functions is equivalent to (24). ■

For general nonlinear functions the convergence analysis of our algorithm exploits insights developed by Gilbert and Nocedal [15], Dai and Liao [9] and that of Hager and Zhang [16]. Global convergence proof of NDHSDY algorithm is based on the Zoutendijk condition combined with the analysis showing that the sufficient descent condition holds and $\|d_k\|$ is bounded. Suppose that the level set S is bounded and the function f is bounded from below. Additionally, assume that there exists a constant $\gamma\geq 0$, such that $\gamma\leq\|g_k\|$.

Theorem 4. *Suppose that the assumptions (i) and (ii) hold and for every $k\geq 0$ there exist the constants $\eta\geq 0$ and $\omega\geq 0$ such that: $\|g_{k+1}\|\leq\eta\|s_k\|$ and $\|g_{k+1}\|\leq\omega\|g_k\|^2/\|s_k\|^2$. If d_k is a descent direction and $\nabla f(x)$ is a Lipschitz function on S, then for the computational scheme (2) and (8)-(9), where $0<c_1\leq\theta_k<1$ and α_k determined by the Wolfe line search (4)-(5) is bounded, either $g_k=0$ for some k or*

$$\liminf_{k\to\infty}\|g_k\|=0. \qquad (26)$$

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

$$\left|\beta_k^C\right| \le \left|\frac{y_k^T g_{k+1}}{y_k^T s_k}\right| + \left|\frac{g_{k+1}^T g_{k+1}}{y_k^T s_k}\right| \le \frac{\|g_{k+1}\|}{\left|y_k^T s_k\right|}\left[\|y_k\|+\|g_{k+1}\|\right]. \tag{27}$$

By the Wolfe condition (5) we have:

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

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

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

On the other hand, from Lipschitz continuity we have $\|y_k\| = \|g_{k+1}-g_k\| \le L\|s_k\|.$ With these, from (27) we get

$$\left|\beta_k^C\right| \le \frac{\omega}{(1-\sigma)\delta}\frac{1}{\|s_k\|^2}\left[L\|s_k\|+\eta\|s_k\|\right] = \frac{\omega(L+\eta)}{(1-\sigma)\delta}\frac{1}{\|s_k\|}. \tag{28}$$

Now, we can write

$$\|d_{k+1}\| \le \|g_{k+1}\| + \left|\beta_k^C\right|\|s_k\| \le \Gamma + \frac{\omega(L+\eta)}{(1-\sigma)\delta}. \tag{29}$$

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

$$0 < \sum_{k=0}^{\infty}\frac{(g_k^T d_k)^2}{\|d_k\|^2} < +\infty , \tag{30}$$

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

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

which contradicts (29). Hence, $\gamma = \liminf_{k\to\infty}\|g_k\| = 0.$ ■

Numerical experiments. In this section we present the computational performance of a Fortran implementation of the NDHSDY algorithm on a set of 750 unconstrained optimization test problems. The test problems are the unconstrained problems in the CUTE [7] library, along with other large-scale optimization problems presented in [1]. We selected 75 large-scale unconstrained optimization problems in extended or generalized form. Each problem is tested 10 times for a gradually increasing number of variables: $n=1000,2000,\ldots,10000.$ At the same time we present comparisons with other conjugate gradient algorithms, including the performance profiles of Dolan and Moré [12]. All algorithms implement the Wolfe line search conditions with $\rho=0.0001$ and $\sigma=0.9$, and the same stopping criterion $\|g_k\|_\infty \le 10^{-6}$, where $\|.\|_\infty$ is the maximum absolute component of a vector. The comparisons of algorithms are given in the following context. Let f_i^{ALG1} and f_i^{ALG2} be the optimal value found by ALG1 and ALG2, for problem $i=1,\ldots,750,$ respectively. We say that, in the particular problem $i,$ the performance of ALG1 was better than the performance of ALG2 if:

$$\left| f_i^{ALG1} - f_i^{ALG2} \right| < 10^{-3} \tag{31}$$

and the number of iterations, or the number of function-gradient evaluations, or the CPU time of ALG1 was less than the number of iterations, or the number of function-gradient evaluations, or the CPU time corresponding to ALG2, respectively. In this numerical study we declare that a method solved a particular problem if the final point obtained has the lowest functional value among the tested methods (up to 10^{-3} tolerance as it is specified in (31)). Clearly, this criterion is acceptable for users that are interested in minimizing functions and not finding critical points. All codes are written in double precision Fortran and compiled with f77 (default compiler settings) on an Intel Pentium 4, 1.8GHz workstation. All these codes are authored by Andrei.

In the first set of numerical experiments we compare the performance of NDHSDY to the HS and DY conjugate gradient algorithms. Figure 1 presents the Dolan and Moré CPU performance profiles of NDHSDY versus HS and DY, respectively.

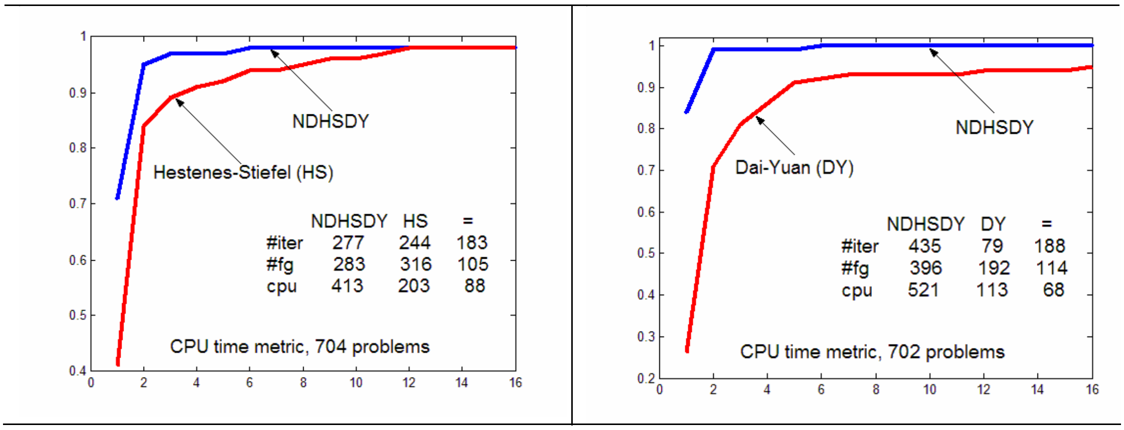
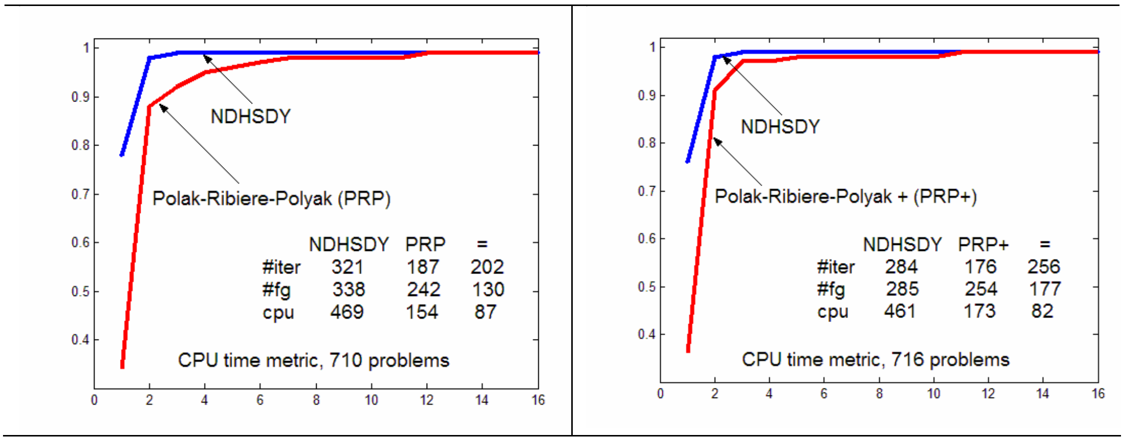


Fig. 1. Performance based on CPU time. NDHSDY versus HS and DY.

When comparing NDHSDY to HS, subject to the number of iterations, we see that NDHSDY was better in 277 problems (i.e. it achieved the minimum number of iterations in 277 problems), HS was better in 244 problems and they achieved the same number of iterations in 183 problems, etc. Out of 750 problems, only for 704 problems the criterion (31) holds. Similarly, we see the number of problems for which NDHSDY was better than DY. Observe that the convex combination of HS and DY, expressed as in (7), is far more successful than HS or DY algorithms.

Figure 2 presents the performance profiles of NDHSDY versus the conjugate gradient algorithms: PRP, PRP+, LS and CD. It seems that the best algorithm is the hybrid algorithm NDHSDY given by a convex combination of HS and DY, where the parameter in the convex combination is obtained using the Newton direction.



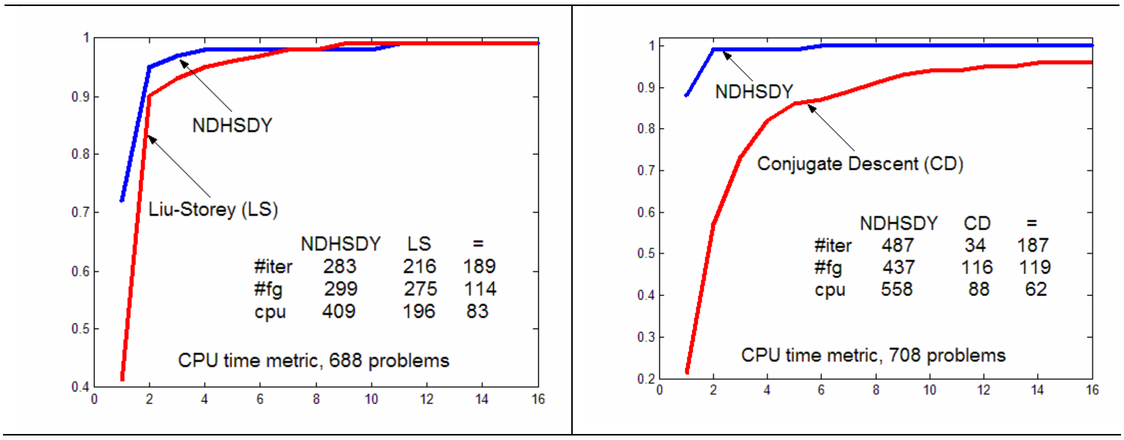


Fig. 2. Performance profiles of NDHSDY versus some conjugate gradient algorithms.

Observe that the NDHSDY algorithm is top performer. Since these codes use the same Wolfe line search and the same stopping criterion they differ in their choice of the search direction. Hence, among these hybrid conjugate gradient algorithms we considered here, NDHSDY appears to generate the best search direction. Also, the algorithm has better performance profiles than those corresponding to HS and DY. In this numerical study we noticed that for most of the iterations the NDHSDY algorithm uses β_k^C. Referring to the condition (10) we noticed that $(y_k^T g_{k+1})(s_k^T g_{k+1})/y_k^T s_k$ tends to zero faster than $\|g_{k+1}\|^2$. For most of the iterations the condition (10) is satisfied, i.e. the algorithm has a self-adjusting property in the sense given in [8]. It is worth saying that the condition (10) is more satisfied after those iterations in which β_k^C is computed according to the HS or DY rules. Introducing (10) as a restart criterion, does not improve the performances of the algorithm. On the other hand, the conditions $\|g_{k+1}\| \le \eta \|s_k\|$ and $\|g_{k+1}\| \le \omega \|g_k\|^2 / \|s_k\|^2$ from theorem 4 say that $\|g_{k+1}\|^3 \le \omega\eta^2 \|g_k\|^2$. We noticed that there exists a k_0 such that for any iteration $k \ge k_0$ the above condition $\|g_{k+1}\|^3 \le \omega\eta^2 \|g_k\|^2$ is satisfied, illustrating the global convergence.

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

References

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

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

[3] N. Andrei, *Scaled memoryless BFGS preconditioned conjugate gradient algorithm for unconstrained optimization.* Optimization Methods and Software, 22 (2007), 561-571.
[4] N. Andrei, *A scaled BFGS preconditioned conjugate gradient algorithm for unconstrained optimization*. Applied Mathematics Letters, 20 (2007), 645-650.
[5] N. Andrei, *Performance profiles of conjugate gradient algorithms for unconstrained optimization.* This issue.
[6] E. Birgin and J.M. Martínez, *A spectral conjugate gradient method for unconstrained optimization*, Applied Math. and Optimization, 43, pp.117-128, 2001.
[7] I. Bongartz, A.R. Conn, N.I.M. Gould and P.L. Toint, *CUTE: constrained and unconstrained testing environments*, ACM Trans. Math. Software, 21, pp.123-160, 1995.
[8] Y.H. Dai, *New properties of a nonlinear conjugate gradient method.* Numer. Math., 89 (2001), pp.83-98.
[9] Y.H. Dai and L.Z. Liao, *New conjugacy conditions and related nonlinear conjugate gradient methods.* Appl. Math. Optim., 43 (2001), pp. 87-101.
[10] Y.H. Dai, Han, J.Y., Liu, G.H., Sun, D.F., Yin, .X. and Yuan, Y., *Convergence properties of nonlinear conjugate gradient methods*. SIAM Journal on Optimization, 10 (1999), 348-358.
[11] Y.H. Dai and Y. Yuan, *A nonlinear conjugate gradient method with a strong global convergence property*, SIAM J. Optim., 10 (1999), pp. 177-182.
[12] E.D. Dolan and J.J. Moré, "*Benchmarking optimization software with performance profiles*", Math. Programming, 91 (2002), pp. 201-213.
[13] R. Fletcher, *Practical Methods of Optimization, vol. 1: Unconstrained Optimization*, John Wiley & Sons, New York, 1987.
[14] R. Fletcher and C. Reeves, *Function minimization by conjugate gradients*, Comput. J., 7 (1964), pp.149-154.
[15] J.C. Gilbert and J. Nocedal, *Global convergence properties of conjugate gradient methods for optimization*, SIAM J. Optim., 2 (1992), pp. 21-42.
[16] W.W. Hager and H. Zhang, "*A new conjugate gradient method with guaranteed descent and an efficient line search*", SIAM Journal on Optimization, 16 (2005) 170-192.
[17] W.W. Hager and H. Zhang, *A survey of nonlinear conjugate gradient methods*. Pacific journal of Optimization, 2 (2006), pp.35-58.
[18] M.R. Hestenes and E.L. Stiefel, *Methods of conjugate gradients for solving linear systems*, J. Research Nat. Bur. Standards, 49 (1952), pp.409-436.
[19] Y. Liu, and C. Storey, *Efficient generalized conjugate gradient algorithms, Part 1: Theory.* JOTA, 69 (1991), pp.129-137.
[20] E. Polak and G. Ribière, *Note sur la convergence de directions conjuguée,* Rev. Francaise Informat Recherche Operationelle, 3e Année 16 (1969), pp.35-43.
[21] B.T. Polyak, *The conjugate gradient method in extreme problems*. USSR Comp. Math. Math. Phys., 9 (1969), pp.94-112.
[22] M.J.D. Powell, *Nonconvex minimization calculations and the conjugate gradient method.* in Numerical Analysis (Dundee, 1983), Lecture Notes in Mathematics, vol. 1066, Springer-Verlag, Berlin, 1984, pp.122-141.
[23] D.F. Shanno and K.H. Phua, *Algorithm 500, Minimization of unconstrained multivariate functions*, ACM Trans. on Math. Soft., 2, pp.87-94, 1976.

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

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

August 12, 2007