New hybrid conjugate gradient algorithms as a convex combination of PRP and DY for unconstrained optimization

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

Abstract. In the hybrid conjugate gradient algorithms, we suggest in this letter, the parameter β_k is computed as a convex combination of the Polak-Ribière-Polyak and Dai-Yuan conjugate gradient algorithms, i.e. $\beta_k = (1-\theta_k)\beta_k^{PRP} + \theta_k \beta_k^{DY}$. In one hybrid algorithm the parameter θ_k is computed in such a way that the conjugacy condition is satisfied, independent of the line search. In the other, it is computed in such a way that the conjugate gradient direction is the Newton direction. The algorithms use the standard Wolfe line search conditions. Numerical comparisons with conjugate gradient algorithms using a set of 750 unconstrained optimization problems, some of them from the CUTE library, show that the hybrid computational scheme based on conjugacy condition outperform the known hybrid conjugate gradient algorithms.

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

1. Introduction

Let us consider the nonlinear unconstrained optimization problem

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

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

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

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

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

In (3) β_k is known as the conjugate gradient parameter, $s_k = x_{k+1} - x_k$ and $g_k = \nabla f(x_k)$. Consider $\|.\|$ 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 [10]. Different conjugate gradient algorithms correspond to different choices for the scalar parameter β_k. Some of these methods as Fletcher and Reeves (FR) [8], Dai and Yuan (DY) [3] and Conjugate Descent (CD) proposed by Fletcher [7]:

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

have strong convergence properties, but they may have modest practical performance due to jamming. On the other hand, the methods of Polak – Ribière [14] and Polyak (PRP) [15], Hestenes and Stiefel (HS) [11] or Liu and Storey (LS) [13]:

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

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

In this paper we focus on hybrid conjugate gradient methods. These algorithms have been devised to exploit the attractive features of the above conjugate gradient algorithms. They are defined by (2) and (3) where the parameter β_k is 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 [4] (hDY)
2.	$\beta_k^{hDYz} = max\left\{0, min\left\{\beta_k^{HS}, \beta_k^{DY}\right\}\right\}$	Hybrid Dai-Yuan zero [4] (hDYz)
3.	$\beta_k^{GN} = max\left\{-\beta_k^{FR}, min\left\{\beta_k^{PRP}, \beta_k^{FR}\right\}\right\}$	Gilbert and Nocedal [9] (GN)
4.	$\beta_k^{HuS} = max\left\{0, min\left\{\beta_k^{PRP}, \beta_k^{FR}\right\}\right\}$	Hu and Storey [12] (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 [16] (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)

In this paper we propose another hybrid conjugate gradient as a convex combination of PRP and DY conjugate gradient algorithms. We selected these two methods to combine in a hybrid conjugate gradient algorithm because PRP has good computational properties, on one side, and DY has strong convergence properties, on the other side. Often PRP method performs better in practice than DY and we speculate this in order to have a good practical conjugate algorithm. The iterates $x_0, x_1, x_2, \ldots$ of our algorithms are computed by means of the recurrence (2) where the stepsize $\alpha_k > 0$ is determined according to the Wolfe conditions (4) and (5), and the directions d_k are generated as:

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

where

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

and θ_k is a scalar parameter satisfying $0 \le \theta_k \le 1$, which follows to be determined. Observe that if $\theta_k = 0$, then $\beta_k^N = \beta_k^{PRP}$, and if $\theta_k = 1$, then $\beta_k^N = \beta_k^{DY}$. On the other hand, if $0 < \theta_k < 1$, then β_k^N is a convex combination of β_k^{PRP} and β_k^{DY}. It easy to see that:

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

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

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

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

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

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

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

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

$$= \frac{g_k^T s_k}{y_k^T s_k}\left\|g_{k+1}\right\|^2 + \frac{y_k^T g_{k+1}}{g_k^T g_k} g_{k+1}^T s_k .$$

But, $y_k^T s_k > 0$ by (5) and since $g_k^T s_k \leq 0$, it follows that $\frac{g_k^T s_k}{y_k^T s_k}\left\|g_{k+1}\right\|^2 \leq 0$. Therefore, from (9), it follows that $g_{k+1}^T d_{k+1} \leq 0$, i.e. the direction d_{k+1} is a descent one. ■

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

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

Proof. From (8) we have:

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

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

Observe that, since $y_k^T s_k > 0$ by (5) and since $g_{k+1}^T s_k = y_k^T s_k + g_k^T s_k < y_k^T s_k$, then $y_k^T s_k / g_{k+1}^T s_k > 1$. Therefore, if $0<\theta_k<1$, it follows that $\theta_k < y_k^T s_k / g_{k+1}^T s_k$. Therefore $1-\theta_k \frac{g_{k+1}^T s_k}{y_k^T s_k} > 0$, proving the theorem. ■

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

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

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

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

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

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

Observe that the parameter θ_k given by (11) or (13) can be outside the interval $[0,1]$. However, in order to have a real convex combination in (7) the following rule is considered: if $\theta_k \le 0$, then set $\theta_k = 0$ in (7), i.e. $\beta_k^N = \beta_k^{PRP}$; if $\theta_k \ge 1$, then take $\theta_k = 1$ in (7), i.e. $\beta_k^N = \beta_k^{DY}$. Therefore, under this rule for θ_k selection, the direction d_{k+1} in (8) combines in a convex combination manner the properties of PRP and DY algorithms.

2. The New Hybrid Conjugate Gradient Algorithms (CCOMB, NDOMB)

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

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

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

Step 4. θ_k parameter computation. If $(y_k^T g_{k+1})(y_k^T s_k) - \|g_{k+1}\|^2 \|g_k\|^2 = 0$, then set $\theta_k = 0$, otherwise compute θ_k as follows:

CCOMB algorithm (θ_k from Conjugacy Condition): $\theta_k = \theta_k^{CCOMB}$.

NDOMB algorithm (θ_k from Newton Direction): $\theta_k = \theta_k^{NDOMB}$.

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

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

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

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

It is well known that if f is bounded along the direction d_k then there exists a stepsize α_k satisfying the Wolfe line search conditions (4) and (5). In our algorithm when the Powell restart condition is satisfied, then we restart the algorithm with the negative gradient $-g_{k+1}$. More sophisticated reasons for restarting the algorithms have been proposed in the literature [5], but we are interested in the performance of a conjugate gradient algorithm that uses this restart criterion, associated to a direction satisfying the conjugacy condition or is equal to the Newton direction. Under reasonable assumptions, conditions (4), (5) and (14) are sufficient to prove the global convergence of the algorithm.

3. Numerical experiments and comparisons

In this section we present the computational performance of a Fortran implementation of the CCOMB and NDOMB algorithms on a set of 750 unconstrained optimization test problems. The test problems are the unconstrained problems in the CUTE [2] 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 [6].

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:

$$\left| f_i^{ALG1} - f_i^{ALG2} \right| < 10^{-3} \tag{15}$$

and the number of iterations, or the number of function-gradient evaluations, or the CPU time of ALG1 was less than the number of iterations, or the number of function-gradient evaluations, or the CPU time corresponding to ALG2, respectively.
All codes are written in double precision Fortran and compiled with f77 (default compiler settings) on an Intel Pentium 4, 1.8GHz workstation. All these codes are authored by Andrei. The performances of these algorithms have been evaluated using the profiles of Dolan and Moré [6]. That is, for each algorithm we plot the fraction of problems for which the algorithm is within a factor of the best CPU time. The left side of these Figures gives the percentage of the test problems, out of 750, for which an algorithm is more successful; the right side gives the percentage of the test problems that were successfully solved by each of the algorithms. Mainly, the right side represents a measure of an algorithm's robustness. Figure 1 shows the performance profiles of CCOMB and NDOMB versus PRP and DY, respectively.

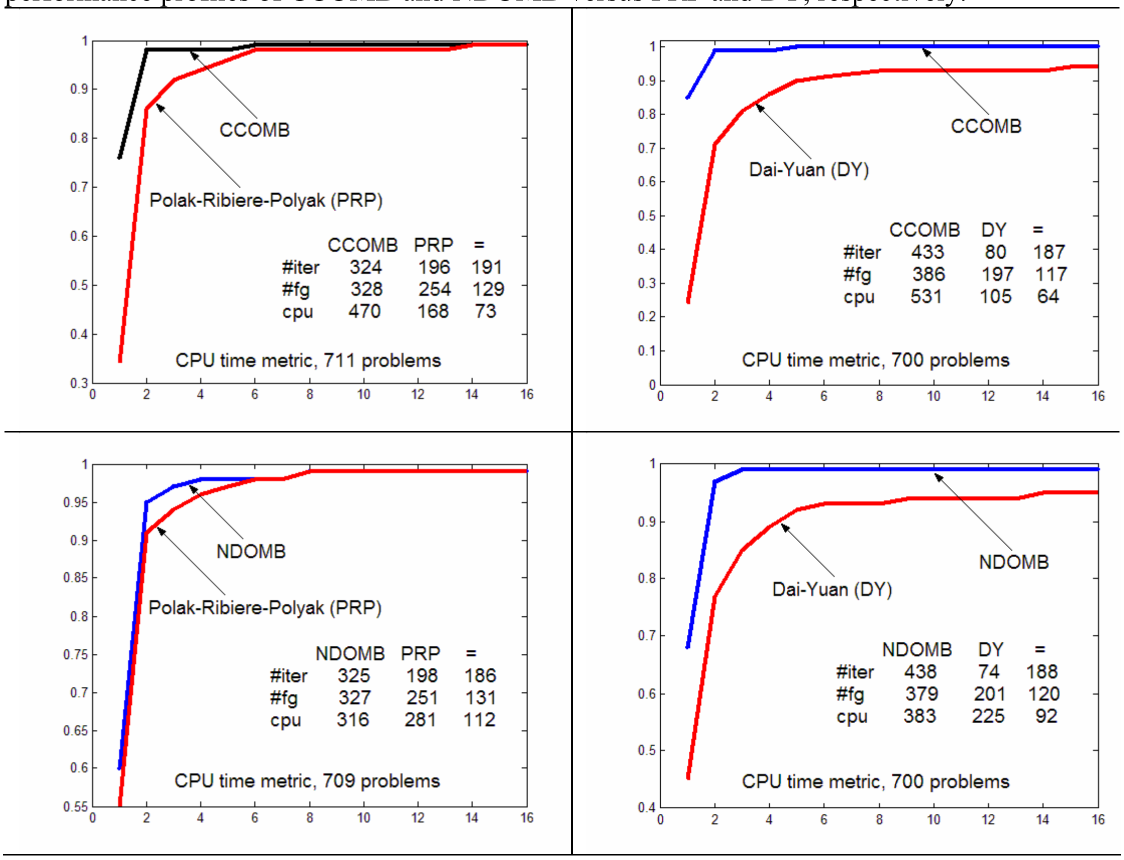

Fig. 1. Performance profiles based on CPU time.

From Figure 1 we see that both CCOMB and NDOMB are more performant than PRP and DY. Observe that, CCOMB is more successful than NDOMB. In Figure 2 we present the Dolan-Moré performance profiles of CCOMB versus some hybrid conjugate gradient algorithms.

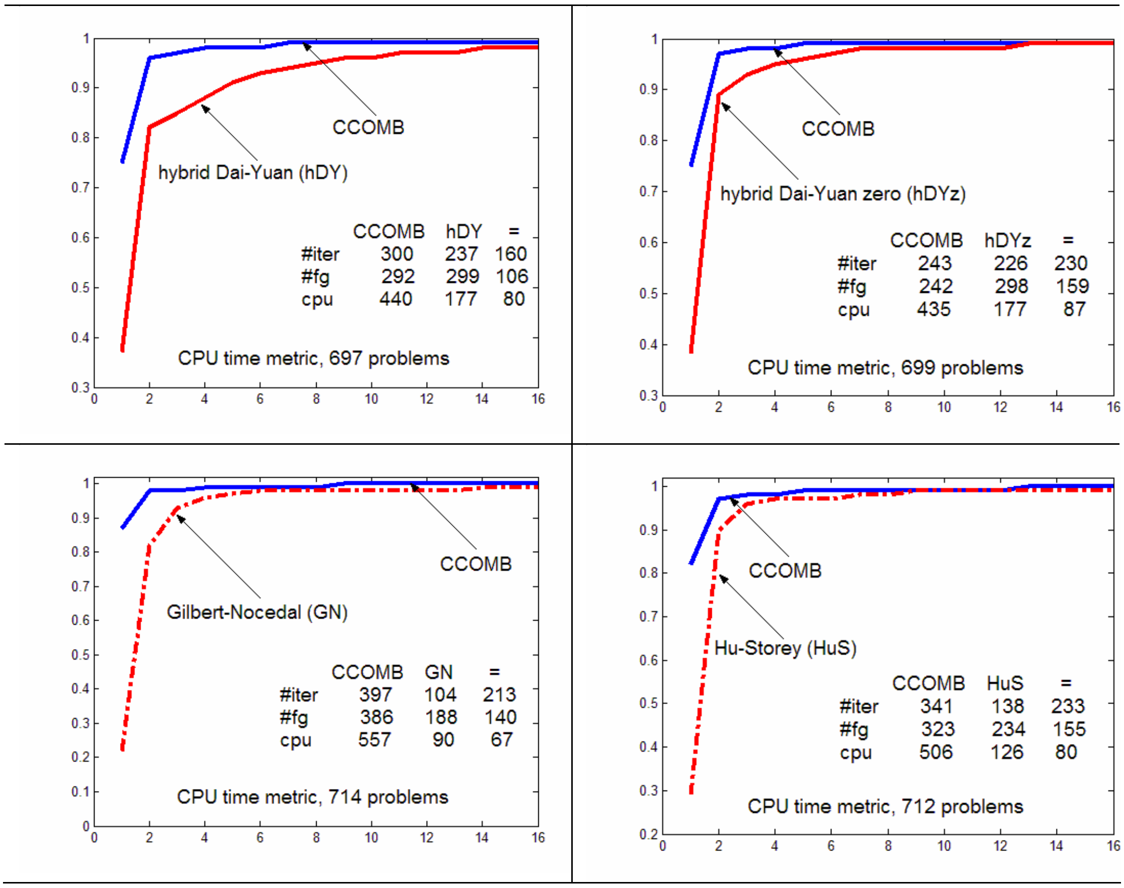

Fig. 2. Performance profiles of CCOMB versus hybrid algorithms: hDY, hDYz, GN and HuS.

4. Conclusion

The known hybrid conjugate gradient algorithms are based on projection of the classical conjugate gradient algorithms FR, DY, CD, PRP, HS and LS. In this paper we have proposed new hybrid conjugate gradient algorithms in which the parameter β_k is computed as a convex of β_k^{PRP} and β_k^{DY}, i.e. $\beta_k = (1-\theta_k)\beta_k^{PRP} + \theta_k \beta_k^{DY}$. The parameter θ_k is computed in such a manner that the conjugacy condition is satisfied, or the corresponding direction in hybrid conjugate gradient algorithm is the Newton direction. The Dolan and Moré CPU performance profile of hybrid conjugate gradient algorithm based on conjugacy condition (CCOMB algorithm) is higher than the performance profile corresponding to the hybrid algorithm based to the Newton direction (NDOMB algorithm). The performance profile of CCOMB algorithm was higher than those of the well established hybrid conjugate gradient algorithms (hDY, hDYz, GN, HuS) for a set consisting of 750 unconstrained optimization test problems, some of them from CUTE library. Additionally the proposed hybrid conjugate gradient algorithm CCOMB is more robust than the PRP and DY conjugate gradient algorithms.

References

[1] N. Andrei, "*Test functions for unconstrained optimization*". http://www.ici.ro / camo / neculai/ SCALCG/ evalfg.for
[2] 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.
[3] 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.
[4] Y.H. Dai and Y. Yuan, *An efficient hybrid conjugate gradient method for unconstrained optimization*, Ann. Oper. Res., 103 (2001), pp. 33-47.

[5] Y.H. Dai, L.Z. Liao and Duan Li, *On restart procedures for the conjugate gradient method.* Numerical Algorithms 35 (2004), pp. 249-260.
[6] E.D. Dolan and J.J. Moré, "*Benchmarking optimization software with performance profiles*", Math. Programming, 91 (2002), pp. 201-213.
[7] R. Fletcher, *Practical Methods of Optimization, vol. 1: Unconstrained Optimization*, John Wiley & Sons, New York, 1987.
[8] R. Fletcher and C. Reeves, *Function minimization by conjugate gradients*, Comput. J., 7 (1964), pp.149-154.
[9] J.C. Gilbert and J. Nocedal, *Global convergence properties of conjugate gradient methods for optimization*, SIAM J. Optim., 2 (1992), pp. 21-42.
[10] W.W. Hager and H. Zhang, *A survey of nonlinear conjugate gradient methods*. Pacific journal of Optimization, 2 (2006), pp.35-58.
[11] 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.
[12] Y.F. Hu and C. Storey, *Global convergence result for conjugate gradient methods.* J. Optim. Theory Appl., 71 (1991), pp.399-405.
[13] Y. Liu, and C. Storey, *Efficient generalized conjugate gradient algorithms, Part 1: Theory.* JOTA, 69 (1991), pp.129-137.
[14] E. Polak and G. Ribière, *Note sur la convergence de directions conjuguée,* Rev. Francaise Informat Recherche Operationelle, 3e Année 16 (1969), pp.35-43.
[15] B.T. Polyak, *The conjugate gradient method in extreme problems*. USSR Comp. Math. Math. Phys., 9 (1969), pp.94-112.
[16] D. Touati-Ahmed and C. Storey, *Efficient hybrid conjugate gradient techniques.* J. Optim. Theory Appl., 64 (1990), pp.379-397.

October 1, 2007
Sent to Rodin (AML)