Yokohama Publishers
ISSN 1349-8169 ONLINE JOURNAL

Pacific Journal of Optimization
Vol. 12, No. 4, pp. 847–859, October 2016

A HYBRID NONLINEAR CONJUGATE GRADIENT METHOD FOR UNCONSTRAINED OPTIMIZATION PROBLEMS

P. Kaelo

Abstract: In this article, we propose a hybrid conjugate gradient method for solving unconstrained optimization problems that possesses the sufficient descent condition. Global convergence, with the weak Wolfe line search conditions, of the proposed hybrid conjugate gradient method is also established. Numerical results of the new hybrid method show that the proposed method is very competitive.

Key words: *hybrid conjugate gradient, weak Wolfe conditions, Global convergence*

Mathematics Subject Classification: *90C06, 90C30, 65K05*

1 Introduction

Consider the unconstrained optimization problem

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

ISSN 1348-9151

where $f : \mathbb{R}^n \to \mathbb{R}$ is a continuously differentiable function. Conjugate gradient method is a very powerful technique for solving problem (1.1), particularly when it is of larger scale. It has advantages over Newton and quasi-Newton methods in that it only needs the first order derivatives and hence less storage capacity is needed. Its program is also relatively simple.

Given an initial guess $x_0 \in \mathbb{R}^n$, the nonlinear conjugate gradient method generates a sequence $\{x_k\}$ for problem (1.1) as

$$x_{k+1} = x_k + \alpha_k d_k, \qquad k = 0, 1, 2, \ldots, \tag{1.2}$$

where α_k is a step length which is determined by a line search and d_k is a descent direction of f at x_k generated as

$$d_k = \begin{cases} -g_k, & \text{if} \quad k = 0, \\ -g_k + \beta_k d_{k-1}, & \text{if} \quad k \geq 1, \end{cases} \tag{1.3}$$

where $g_k = \nabla f(x_k)$ is the gradient of f at x_k and β_k is a scalar.

Different choices of the scalar β_k lead to different conjugate gradient methods, with well known formulas for β_k being Fletcher-Reeves (FR) method [13]

$$\beta_k = \frac{||g_k||^2}{||g_{k-1}||^2},$$

Polak-Ribiére-Polyak (PRP) method [24, 25]

$$\beta_k = \frac{g_k^T y_{k-1}}{||g_{k-1}||^2},$$

Dai-Yuan (DY) method [8]

$$\beta_k = \frac{||g_k||^2}{d_{k-1}^T y_{k-1}},$$

and the Hestenes-Stiefel (HS) method [16]

$$\beta_k = \frac{g_k^T y_{k-1}}{d_{k-1}^T y_{k-1}},$$

where $y_{k-1} = g_k - g_{k-1}$ and $||\cdot||$ denotes the Euclidean norm of vectors. These were the first scalars β_k for nonlinear conjugate gradient methods to be proposed. Since then, other scalars β_k have been proposed in the literature (see for example [4, 10, 12, 31, 33] and references therein).

Hybrid scalars β_k have also been suggested in the literature, where a combination of two or more scalars are used in the same conjugate gradient method. Hybrids try to combine attractive features of different algorithms. For instance, some methods have good global convergence properties whilst they do not perform very well numerically, due to their inability to avoid jamming, and others perform very well numerically but may not always converge globally. Touti-Ahmed and Storey [27] proposed this hybrid method

$$\beta_k = \begin{cases} \beta_k^{PRP}, & \text{if } 0 \leq \beta_k^{PRP} \leq \beta_k^{FR} \\ \beta_k^{FR} & \text{otherwise,} \end{cases}$$

to exploit the attractive convergence properties of β_k^{FR} and to avoid jamming β_k^{PRP} is used, which in turn gives nice numerical performance. One other example of a hybrid that uses the attractive features of β_k^{FR} and β_k^{PRP} is that of Hu and Storey [17]

$$\beta_k = \max\left\{0, \min\left(\beta_k^{FR}, \beta_k^{PRP}\right)\right\}.$$

Other hybrids have been proposed by introducing parameters that combine them. For example, Dai and Yuan [7] introduced a one-parameter family of conjugate gradient methods by proposing

$$\beta_k = \frac{||g_k||^2}{\lambda_k||g_{k-1}||^2 + (1-\lambda_k)d_{k-1}^T y_{k-1}},$$

where $\lambda_k \in [0, 1]$ is a parameter. More examples of hybrid conjugate gradient methods can be found in [1, 3, 5, 9, 14, 15, 18–21, 30, 32].

The step length α_k is often chosen to satisfy certain line search conditions. It is very important in the convergence analysis and implementation of conjugate gradient methods. A number of line search rules have been discussed in the literature, see for example [6, 11, 23, 26, 31]. However, the weak Wolfe conditions

$$f(x_k + \alpha_k d_k) \leq f(x_k) + \delta\alpha_k g_k^T d_k \tag{1.4}$$

and

$$g(x_k + \alpha_k d_k)^T d_k \geq \sigma g_k^T d_k, \tag{1.5}$$

or the strong Wolfe conditions

$$f(x_k + \alpha_k d_k) \leq f(x_k) + \delta \alpha_k g_k^T d_k \tag{1.6}$$

and

$$|g(x_k + \alpha_k d_k)^T d_k| \leq -\sigma g_k^T d_k, \tag{1.7}$$

where $0 < \delta < \sigma < 1$, are often used in the convergence analysis and implementation of nonlinear conjugate gradient methods, see for example [1, 10, 20, 23, 29, 31, 32, 34].

In this paper, we suggest another approach to get a new hybrid nonlinear conjugate gradient method. In section 2, we present the proposed method. In Section 3 we prove that the proposed algorithm (method) globally converges. Section 4 presents some numerical experiments and conclusion is given in Section 5.

2 Description of the Method

In this section, we present our proposed hybrid conjugate gradient method. This technique we are proposing here is motivated by the work of Yueting and Mingyuan [32] and that of Dai and Wen [10]. In [32], a new β_k is proposed as

$$\beta_k = \begin{cases} \dfrac{g_k^T g_k}{\mu |g_k^T d_{k-1}| + d_{k-1}^T y_{k-1}}, & \text{if } ||g_k||^2 \geq |g_k^T g_{k-1}| \\ 0, & \text{otherwise,} \end{cases} \tag{2.1}$$

where $\mu > 1$ is a parameter. The β_k proposed above restarts the algorithm with $d_k = -g_k$ whenever $|g_k^T g_{k-1}| > ||g_k||^2$. This method was shown to perform very well numerically. On the other hand, Dai and Wen [10] proposed a modification to the HS method as

$$\beta_k = \frac{||g_k||^2 - \frac{||g_k||}{||g_{k-1}||} |g_k^T g_{k-1}|}{\mu |g_k^T d_{k-1}| + d_{k-1}^T y_{k-1}}. \tag{2.2}$$

This β_k satisfies $\beta_k \geq 0$ and produces sufficient descent directions with the weak Wolfe conditions. Its global convergence was also established. Now, following the work described above, we herein combine the two β_k parameters to come up with

$$\beta_k = \begin{cases} \dfrac{g_k^T \left(g_k - \frac{||g_k||}{||g_{k-1}||} g_{k-1} \right)}{\mu |d_{k-1}^T g_k| + d_{k-1}^T y_{k-1}}, & 0 \leq g_k^T g_{k-1} \leq ||g_k||^2 \\ \dfrac{g_k^T g_k}{\mu |d_{k-1}^T g_k| + d_{k-1}^T y_{k-1}}, & \text{otherwise,} \end{cases} \tag{2.3}$$

where $\mu > 1$ is a constant. It is clear from the definition of β_k above that

$$\begin{aligned} 0 &= \frac{||g_k||^2 - \frac{||g_k||}{||g_{k-1}||} ||g_k|| ||g_{k-1}||}{\mu |d_{k-1}^T g_k| + d_{k-1}^T y_{k-1}} \\ &\leq \frac{g_k^T \left(g_k - \frac{||g_k||}{||g_{k-1}||} g_{k-1} \right)}{\mu |d_{k-1}^T g_k| + d_{k-1}^T y_{k-1}} \\ &\leq \frac{g_k^T g_k}{\mu |d_{k-1}^T g_k| + d_{k-1}^T y_{k-1}}, \end{aligned}$$

for all $k \geq 1$. Thus,

$$0 \leq \beta_k \leq \frac{g_k^T g_k}{\mu |d_{k-1}^T g_k| + d_{k-1}^T y_{k-1}}, \forall k \geq 1. \tag{2.4}$$

We now present our modified hybrid conjugate gradient method as

Algorithm 2.1. Modified Hybrid Conjugate Gradient Method

1: Give initial guess $x_0 \in \mathbb{R}^n$, $\mu > 1$, $0 < \delta < \sigma < 1$ and the tolerance $\epsilon > 0$.
2: Set $d_0 = -g_0$ and $k = 0$. If $||g_0|| < \epsilon$ then stop.
3: **for** $k = 0, 1, \ldots$ **do**
4: Compute α_k using the weak Wolfe conditions (1.4) and (1.5).
5: Set $x_{k+1} = x_k + \alpha_k d_k$, $k = k + 1$.
6: If $||g_k|| < \epsilon$ stop.
7: Compute β_k using (2.3).
8: Compute $d_k = -g_k + \beta_k d_{k-1}$, go to Step 4.
9: **end for**

The search direction d_k is generally required to satisfy the descent condition

$$g_k^T d_k < 0.$$

However, in order to guarantee convergence, it is often required that it satisfies the sufficient descent condition

$$g_k^T d_k \leq -c||g_k||^2, \tag{2.5}$$

where $c > 0$ is a constant. The descent (or sufficient descent) property is very important for an iterative method to be globally convergent.

Notice that from (1.5), we have that

$$\begin{aligned} d_{k-1}^T y_{k-1} &= d_{k-1}^T (g_k - g_{k-1}) \\ &\geq \sigma d_{k-1}^T g_{k-1} - d_{k-1}^T g_{k-1} \\ &= (\sigma - 1) d_{k-1}^T g_{k-1} \\ &> 0, \forall k \geq 1. \end{aligned} \tag{2.6}$$

Lemma 2.2. *Let $\{d_k, \alpha_k, x_k\}$ be generated by Algorithm 2.1. If $\mu > 1$, then d_k satisfies the sufficient descent condition (2.5) and*

$$g_k^T d_k \leq -\Big(1 - \frac{1}{\mu}\Big)||g_k||^2. \tag{2.7}$$

Proof. For $k = 0$ we have that

$$g_0^T d_0 = -||g_0||^2 \leq -\Big(1 - \frac{1}{\mu}\Big)||g_k||^2,$$

so (2.7) holds. Now we prove that (2.7) holds for $k \geq 0$. From (1.3), (2.3), (2.4) and (2.6),

we have that

$$\begin{aligned} g_k^T d_k &= -g_k^T g_k + \beta_k g_k^T d_{k-1} \\ &\leq -||g_k||^2 + \frac{||g_k||^2}{\mu|g_k^T d_{k-1}| + d_{k-1}^T y_{k-1}} g_k^T d_{k-1} \\ &\leq -||g_k||^2 + \frac{||g_k||^2}{\mu|g_k^T d_{k-1}| + d_{k-1}^T y_{k-1}} |g_k^T d_{k-1}| \\ &\leq -||g_k||^2 + \frac{||g_k||^2}{\mu|g_k^T d_{k-1}|} |g_k^T d_{k-1}| \\ &= -(1 - \frac{1}{\mu})||g_k||^2. \end{aligned}$$

Let $c = (1 - \frac{1}{\mu})$, then (2.5) holds for all $k \geq 0$. □

Now, from (2.4), (2.6) and Lemma 2.2, we obtain that

$$\begin{aligned} g_k^T d_k &= -||g_k||^2 + \beta_k g_k^T d_{k-1} \\ &\leq -||g_k||^2 + \frac{||g_k||^2}{\mu|d_{k-1}^T g_k| + d_{k-1}^T y_{k-1}} g_k^T d_{k-1} \\ &= (-\mu|d_{k-1}^T g_k| + d_{k-1}^T g_{k-1}) \frac{||g_k||^2}{\mu|d_{k-1}^T g_k| + d_{k-1}^T y_{k-1}} \\ &\leq d_{k-1}^T g_{k-1} \frac{||g_k||^2}{\mu|d_{k-1}^T g_k| + d_{k-1}^T y_{k-1}}, \end{aligned}$$

which leads to the condition

$$\frac{||g_k||^2}{\mu|d_{k-1}^T g_k| + d_{k-1}^T y_{k-1}} \leq \frac{d_k^T g_k}{d_{k-1}^T g_{k-1}}. \tag{2.8}$$

Hence, it follows from (2.4) and (2.8) that

$$\beta_k \leq \frac{d_k^T g_k}{d_{k-1}^T g_{k-1}}, \forall k \geq 1. \tag{2.9}$$

3 Global Convergence of the Proposed Method

For the global convergence analysis of the proposed algorithm (Algorithm 2.1), we assume the following assumptions on the objective function hold.

Assumption 3.1. *The function $f(x)$ is bounded from below in the level set*

$$\mathcal{L} = \{x \in \mathbb{R}^n : f(x) \leq f(x_0)\},$$

where x_0 is the starting point.

Assumption 3.2. *In a neighborhood $\mathcal{N}$ of the level set $\mathcal{L}$, $f(x)$ is differentiable and its gradient $g(x)$ is Lipschitz continuous, that is, there exists a constant $L > 0$ such that*

$$||g(x) - g(y)|| \leq L||x - y||, \qquad \forall x, y \in \mathcal{N}.$$

Lemma 3.3. *Suppose Assumptions 3.1 and 3.2 hold. Let x_k be given by (1.2) and (1.3) where d_k is a descent direction, that is, $d_k^T g_k < 0$, and let α_k be determined by the weak Wolfe conditions (1.4) and (1.5), then the Zoutendijk condition [23]*

$$\sum_{k=0}^{\infty} \frac{(g_k^T d_k)^2}{||d_k||^2} < +\infty \tag{3.1}$$

holds.

Proof. From (1.5) and Assumption 3.2, we get that

$$\begin{aligned} -(1-\sigma)d_k^T g_k \leq d_k^T(g_{k+1} - g_k) &\leq ||d_k||||g_{k+1} - g_k|| \\ &\leq L\alpha_k||d_k||^2, \end{aligned}$$

which implies that

$$\alpha_k \geq \frac{(\sigma - 1)}{L||d_k||^2} d_k^T g_k > 0. \tag{3.2}$$

From (1.4) and the inequalty (3.2) above, we obtain

$$f(x_k) - f(x_k + \alpha_k d_k) \geq \frac{\delta(1-\sigma)}{L} \frac{(d_k^T g_k)^2}{||d_k||^2}. \tag{3.3}$$

Using Assumption 3.1 and (3.3), we get that

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

holds. □

Making use of Lemma 3.3, we are now in a position to establish the global convergence of Algorithm 2.1.

Theorem 3.4 (Global convergence)**.** *Suppose that Assumptions 3.1 and 3.2 hold. Let $\{d_k, \alpha_k, x_k\}$ be generated by Algorithm 2.1, $\mu > 1$ and α_k is determined by the weak Wolfe conditions (1.4) and (1.5), then*

$$\lim_{k \to \infty} \inf ||g_k|| = 0. \tag{3.4}$$

Proof. We prove this theorem by contradiction. Suppose the conclusion does not hold. Then there exists a real number $\gamma > 0$ such that $||g_k|| > \gamma$, $\forall k \geq 1$. Since $d_k + g_k = \beta_k d_{k-1}$, we have that

$$\begin{aligned} ||d_k + g_k||^2 &= \beta_k^2 ||d_{k-1}||^2 \\ \Rightarrow ||d_k||^2 + 2g_k^T d_k + ||g_k||^2 &= \beta_k^2 ||d_{k-1}||^2, \\ \Rightarrow ||d_k||^2 &= \beta_k^2 ||d_{k-1}||^2 - ||g_k||^2 - 2g_k^T d_k. \end{aligned} \tag{3.5}$$

Dividing (3.5) on both sides by $(g_k^T d_k)^2$, we get,

$$\begin{aligned}
\frac{||d_k||^2}{(g_k^T d_k)^2} &= \beta_k^2 \frac{||d_{k-1}||^2}{(g_k^T d_k)^2} - \frac{||g_k||^2}{(g_k^T d_k)^2} - 2\frac{g_k^T d_k}{(g_k^T d_k)^2} \\
&\leq \frac{||d_{k-1}||^2}{(g_{k-1}^T d_{k-1})^2} - \frac{||g_k||^2}{(g_k^T d_k)^2} - 2\frac{g_k^T d_k}{(g_k^T d_k)^2} \\
&= \frac{||d_{k-1}||^2}{(g_{k-1}^T d_{k-1})^2} - \frac{||g_k||^2}{(g_k^T d_k)^2} - \frac{2}{(g_k^T d_k)} \\
&= \frac{||d_{k-1}||^2}{(g_{k-1}^T d_{k-1})^2} - \left(\frac{||g_k||}{(g_k^T d_k)} + \frac{1}{||g_k||}\right)^2 + \frac{1}{||g_k||^2} \\
&\leq \frac{||d_{k-1}||^2}{(g_{k-1}^T d_{k-1})^2} + \frac{1}{||g_k||^2} \\
&\leq \frac{||d_{k-1}||^2}{(g_{k-1}^T d_{k-1})^2} + \frac{1}{\gamma^2}.
\end{aligned}$$

From $d_0 = -g_0$, it follows that

$$\frac{||d_k||^2}{(g_k^T d_k)^2} \leq \frac{k+1}{\gamma^2} \tag{3.6}$$

and hence

$$\sum_{k=0}^{\infty} \frac{(g_k^T d_k)^2}{||d_k||^2} \geq \sum_{k=0}^{\infty} \frac{\gamma^2}{k+1} = \infty. \tag{3.7}$$

The above conclusion contradicts Lemma 3.3, thus, (3.4) holds. □

4 Numerical Experiments

In this section, we present some numerical experiments on some test problems chosen from Morè, et al. [22] and Andrei [2] to analyse the efficiency and effectiveness of our proposed hybrid conjugate gradient method. The test problems from these references are widely used in the literature for testing unconstrained optimization algorithms. The test problems selected for testing in this article are presented in Table 1, where the columns 'Prob' and 'Dim' represent the name and dimension of the test problem, respectively, with dimensions of the problems ranging from 2 to 10000.

We compare our proposed new hybrid conjugate gradient method (β_k^{NM}) with the hybrids of Touati-Ahmed and Storey (β_k^{TS}) [27], Liu (β_k^{CY}) [21], Yueting and Mingyuan (β_k^{YM}) [32] and Hu and Storey (β_k^{HS}) [17]. Since β_k^{NM} is a hybrid conjugate gradient method made up of two other parameters, here we also make a comparison with its components. These are

$$\beta_k^{CG1} = \frac{g_k^T \left(g_k - \frac{||g_k||}{||g_{k-1}||} g_{k-1}\right)}{\mu |d_{k-1}^T g_k| + d_{k-1}^T y_{k-1}},$$

$$\beta_k^{CG2} = \begin{cases} \dfrac{g_k^T \left(g_k - \frac{||g_k||}{||g_{k-1}||} g_{k-1}\right)}{\mu |d_{k-1}^T g_k| + d_{k-1}^T y_{k-1}}, & \text{if } g_k^T g_{k-1} \geq 0 \\ 0, & \text{otherwise} \end{cases}$$

Prob	Dim	Prob	Dim
Rosenbrock	2	Biggs EXP6	6
Freudenstein and Roth	2	Osborne 2	11
Beale	2	Broyden tridiagonal	30
Helical valley	3	Trigonometric	1000
Bard	3	Ext. Rosenbrock	5000
			10000
Gaussian	3	Ext. Powell singular	1000
			5000
			10000
Gulf	3	Raydan 2	5000
			10000
Box	3	Ext. Beale	1000
			2000
Powell Singular	4	Ext. Himmelblau	1000
			2000
Wood	4	Brown and Denis	4

Table 1: Table of test problems

and

$$\beta_k^{CG3} = \frac{g_k^T g_k}{\mu |d_{k-1}^T g_k| + d_{k-1}^T y_{k-1}}.$$

Notice here that β_k^{CG3} is just a derivative of β_k^{YM}.

We considered the stopping condition for the methods as $\epsilon = 10^{-5}$, that is, the algorithms were stopped once the condition $||g_k|| < 10^{-5}$ was satisfied. For the algorithms β_k^{TS} and β_k^{HS}, the strong Wolfe conditions (1.6) and (1.7) were used to find the step length α_k, with $\delta = 0.0001$ and $\sigma = 0.33$. For all other algorithms we used the weak Wolfe conditions (1.4) and (1.5) with $\delta = 0.0001$ and $\sigma = 0.9$. In all cases, $\alpha_k = 1$ is always tried first. The parameter $\mu = 1.5$ was found to give overall better results and hence it was used for all the algorithms. All the methods were coded in MATLAB R2015a. Numerical results are compared based on number of iterations, function evaluations and CPU time.

In order to compare and evaluate the performance of our methods, we use the performance profiles tool proposed by Dolan and Moré [11]. This tool evaluates and compares the performance of the set of solvers (methods) S on a set P of test problems. If we assume that there exists n_s solvers and n_p problems, then for each problem $p \in P$ and solver $s \in S$, we define

$$t_{p,s} = \text{ function evaluations required to solve problem } p \text{ by solver } s.$$

We then compare the performance on problem p by solver s with the best performance by any solver on this problem, that is, using the ratio,

$$r(p,s) = \frac{t_{p,s}}{\min\{t_{p,s} : s \in S\}}.$$

In the case when solver s fails to solve problem p, the ratio $r_{p,s}$ is set to some sufficiently large number. The overall performance profile function is then given as

$$\rho_s(\tau) = \frac{1}{n_p} \text{size}\{p : 1 \leq p \leq n_p, \log(r_{p,s}) \leq \tau\},$$

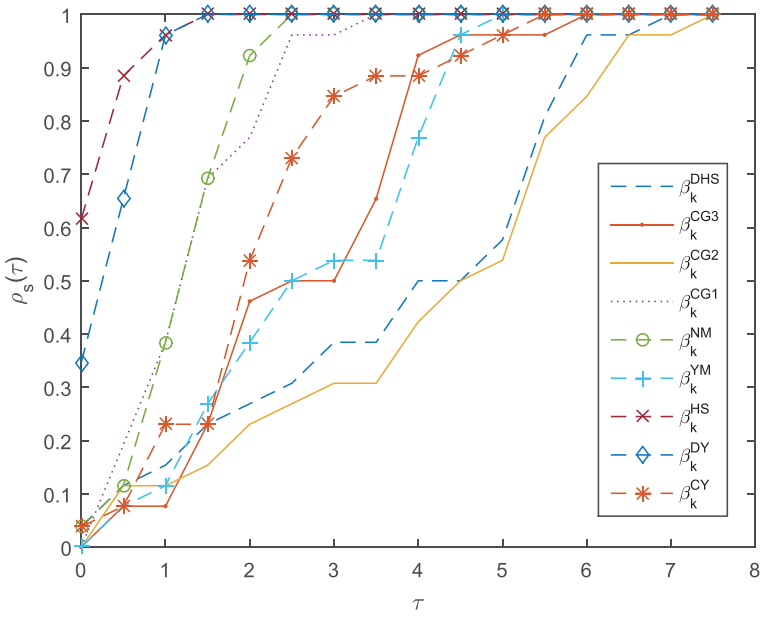

Figure 1: Function evaluations performance profile

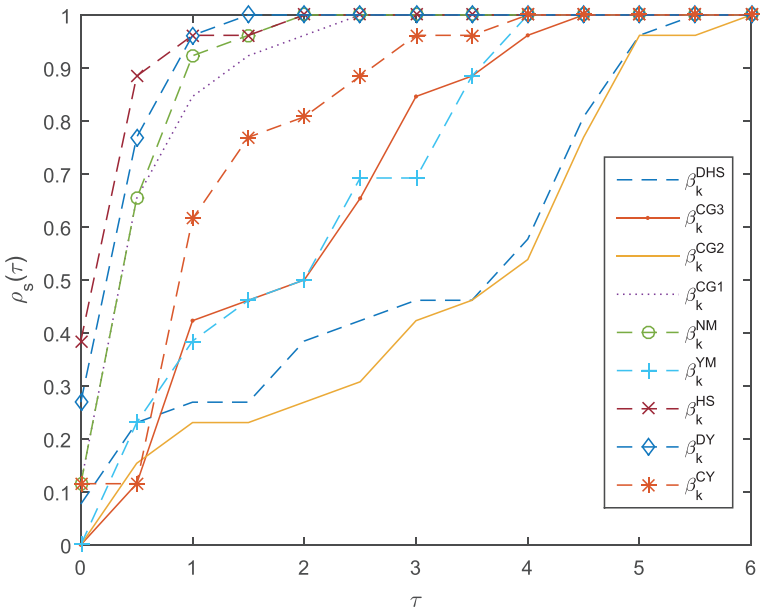

Figure 2: Gradient evaluations performance profile

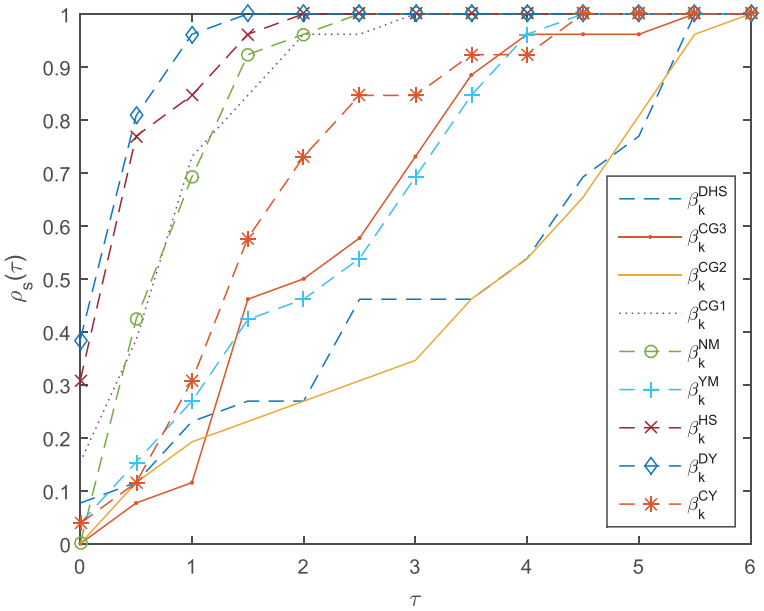

Figure 3: CPU time performance profile

where $\tau \geq 0$. Note here that the function $\rho_s(\tau)$ is such that $\rho_s(\tau) \in [0,1]$ and that the inequality $\rho_s(\tau_1) < \rho_t(\tau_1)$ shows that solver t outperforms solver s at τ_1.

We now plot the performance profiles based on function evaluations, gradient evaluations and CPU time. The results are presented in Figures 1, 2 and 3 for the number of function evaluations, gradient evaluations and CPU time, respectively. We see from Figure 1 that β_k^{HS} and β_k^{TS} outperforms the other methods with our proposed β_k^{NM} being the third best and outperforming all the remaining methods. β_k^{CG2} and β_k^{YM} performed very poorly compared to the rest while β_k^{CG1} is very competitive with β_k^{NM}. We see from the results of β_k^{CG1} and β_k^{CG2} that β_k^{CG1} influences β_k^{NM} more than β_k^{CG2} and this seems to indicate that $g_k^T g_{k-1}$ needs not be too positive.

Figure 2 shows that in terms of gradient evaluations, β_k^{NM} is very competitive with β_k^{HS} and β_k^{TS}, with β_k^{CG1} again being very close to β_k^{NM}, and β_k^{CG2} and β_k^{YM} again performing poorly. The same trend is observed in Figure 3 where again β_k^{NM} is very competitive with β_k^{HS} and β_k^{TS} but outperforms the rest of the other methods. Thus, the performance profiles of these algorithms show that the new method is very competitive and promising.

5 Conclusion

In this work, a new hybrid nonlinear conjugate gradient method for solving unconstrained optimization problems was proposed. This new hybrid conjugate gradient method was shown to possess the sufficient descent condition and also proved to converge globally with the weak Wolfe line search conditions. The new hybrid conjugate gradient method was tested on a number of unconstrained optimization problems that have been extensively used in the literature to test optimization algorithms and the results show that the new method is quite competitive.

Acknowledgment

The author is grateful to the Editor and the two anonymous referees for their constructive comments on this paper, which have greatly improved its presentation. Their comments essentially improved the paper.

References

[1] A.Y. Al-Bayati and W.H. Sharif, A new three term nonlinear conjugate gradient method for unconstrained optimization, *Canad. J. Sci. Eng. Math.* 1 (2010) 108–124.

[2] N. Andrei, An unconstrained optimization test functions collection, *Adv. Model. Optim.* 10 (2008) 147–161.

[3] N. Andrei, Hybrid conjugate gradient algorithm for unconstrained optimization, *J. Optim. Theory Appl.* 141 (2009) 249–264.

[4] N. Andrei, Another nonlinear conjugate gradient algorithm for unconstrained optimization, *Optim. Methods Softw.* 24 (2009) 89–104.

[5] S. Babaie-Kafaki, M. Fatemi and N. Mahdavi-Amiri, Two effective hybrid conjugate gradient algorithms based on modified BFGS updates, *Numer. Algorithms* 58 (2011) 315–331.

[6] Y.-H. Dai, Conjugate gradient methods with Armijo-type line searches, *Acta Math. Appl. Sin. Engl. Ser.* 18 (2002) 123–130.

[7] Y.-H. Dai and Y. Yuan, A class of globally convergent conjugate gradient methods, *Sci. China Ser. A* 46 (2003) 251–261.

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

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

[10] Z. Dai and F. Wen, Another improved Wei-Yao-Liu nonlinear conjugate gradient method with sufficient descent property, *Appl. Math. Comput.* 218 (2012) 7421–7430.

[11] E.D. Dolan and J.J. Moré, Benchmarking optimization software with profile performance profiles, *Math. Program.* 91 (2002) 201–213.

[12] Y. Dong, A practical PR+ conjugate gradient method only using gradient, *Appl. Math. Comput.* 219 (2012) 2041–2052.

[13] R. Fletcher and C. Reeves, Function minimization by conjugate gradients, *Comp. J.* 7 (1964) 149–154.

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

[15] W.W. Hager and H. Zhang, A survey of nonlinear conjugate gradient methods, *Pac. J. Optim.* 2 (2006) 35–58.

[16] M.R. Hestenes and E. Stiefel, Methods of conjugate gradients for solving linear systems, *J. Res. Natl. Bur. Stand.* 49 (1952) 409–436.

[17] Y.F. Hu and C. Storey, Global convergence result of conjugate gradient methods, *J. Optim. Theory Appl.* 71 (1991) 399–405.

[18] H. Iiduka and Y. Narushima, Conjugate gradient methods using value of objective function for unconstrained optimization, *Optim. Lett.* 6 (2012) 941–955.

[19] W. Jia, J. Zong and X. Wang, An improved mixed conjugate gradient method, *Systems Engineering Procedia* 2 (2012) 219–225.

[20] H. Liu, A mixture conjugate gradient method for unconstrained optimization, in: *Third International Symposium on Intelligent Information Technology and Security Informatics*, IEEE, 2010, pp. 26–29.

[21] J. Liu, A hybrid nonlinear conjugate gradient method, Lobachevskii *J. Math.* 33 (2012) 195–199.

[22] J.J. Moré, B.S. Garbow and K.E. Hillstrom, Testing unconstrained optimization software, *ACM Trans. Math. Software* 7 (1981) 17–41.

[23] J. Nocedal and S.J. Wright, Numerical Optimization, 2nd Edition, Springer Science+ Business Media, LLC. Printed in the United States of America, 2006.

[24] E. Polak and G. Ribiére, Note sur la convergence de méthodes de directions conjuguées, *Rev. française Infomat. Recherche Opérationnelle* 3 (1969) 35–43.

[25] B.T. Polyak, The conjugate gradient method in Extreme problems, *USSR Comp. Math. Math. Phys.* 9 (1969) 94–112.

[26] Z.-J. Shi, Convergence of line search methods for unconstrained optimization, *Appl. Math. Comput.* 157 (2004) 393–405.

[27] D. Touati-Ahmed and C. Storey, Efficient hybrid conjugate gradient techniques, *J. Optim. Theory Appl.* 64 (1990) 379–397.

[28] Z. Wei, S. Yao and L. Liu, The convergence properties of some new conjugate gradient methods, *Appl. Math. Comput.* 183 (2006) 1341–1350.

[29] H. Yabe and N. Sakaiwa, A new nonlinear conjugate gradient method for unconstrained optimization, *J. Oper. Res.* 48 (2005) 284–296.

[30] H. Yan, L. Chen and B. Jiao, HS-LS-CD hybrid conjugate gradient algorithm for unconstrained optimization in: *Second International Workshop on Computer Science and Engineering*, IEEE, 2009, pp. 264–268.

[31] G. Yuan and X. Lu, A modified PRP conjugate gradient method, *Ann. Oper. Res.* 166 (2009) 73–90.

[32] Y. Yueting and C. Mingyuan, The global convergence of a new mixed conjugate gradient method for unconstrained optimization, *J. Appl. Math.* 2012, Article ID 932980, http://dx.doi.org/10.1155/2012/932980, 14 pages, 2012.

[33] L. Zhang, W. Zhou and D.-H. Li, A descent modified Polak-Ribiére-Polyak conjugate gradient method and its global convergence, *IMA J. Numer. Anal.* 26 (2006) 629–640.

[34] A. Zhou, Z. Zhu, H. Fan and Q. Qing, Three new hybrid conjugate gradient methods for optimization, *Appl. Math.* 2 (2011) 303–308.

Manuscript received 10 February 2015
revised 13 May 2015
accepted for publication 4 June 2015

P. Kaelo
Department of Mathematics, University of Botswana,
Private Bag UB00704, Gaborone, Botswana
E-mail address: kaelop@mopipi.ub.bw