A Convex Combination of PRP and RMIL methods for large-scale unconstrained optimization problems

GHANIA HADJI
Badji Mokhtar University,Annaba
Mohamed Cherif Messaadia University,Souk Ahras
Annaba, 23000 ; Souk Ahras, 41000
Algeria
ghania.hadji69@gmail.com

YAMINA LASKRI
Superior School of Industrial Technologies
Laboratory LANOS Badji Mokhtar University
Annaba, 23000
Algeria
yamina.laskri@univ-annaba.org

RACHID BENZINE
Superior School of Industrial Technologies
Laboratory LANOS Badji Mokhtar University
Annaba, 23000
Algeria
rabenzine@yahoo.fr

MOHAMMED BELLOUFI
Mohamed Cherif Messaadia University
Laboratory Informatics and Mathematics (LiM)
Souk Ahras , 41000
Algeria
m.belloufi@univ-soukahras.dz

Abstract: The Conjugate Gradient (CG) method is a powerful iterative approach for solving large-scale minimization problems, characterized by its simplicity, low computation cost and good convergence. In this paper, a new hybrid conjugate gradient HLB method (HLB: Hadji-Laskri-Benzine) is proposed and analysed for unconstrained optimization. We compute the parameter β_k^{HLB} as a convex combination of the Polak-Ribière-Polyak $\left(\beta_k^{PRP}\right)$ [1] and the Mohd Rivaie-Mustafa Mamatand Abdelrhaman Abashar $\left(\beta_k^{RMIL+}\right)$ i.e $\beta_k^{HLB} = (1-\theta_k)\,\beta_k^{PRP} + \theta_k \beta_k^{RMIL+}$. By comparing numerically CGHLB with PRP and RMIL+ and by using the Dolan and More CPU performance, we deduce that CGHLB is more efficient.

Key–Words: Unconstrained optimization, hybrid conjugate gradient method, line search, descent property, global convergence.

1 Introduction

Consider the nonlinear unconstrained optimization problem

$$\min_{x\in\mathbb{R}^n} f(x) \tag{1}$$

Where $f:\mathbb{R}^n \to \mathbb{R}$ is a continuously differentiable function, bounded from below. The gradient of f is denoted by $g(x)$. To solve this problem, we start from an initial point $x_0 \in \mathbb{R}^n$. Nonlinear conjugate gradient methods generate sequences $\{x_k\}$ of the following form:

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

where x_k is the current iterate point and $\alpha_k > 0$ is the step size which is obtained by line search.

The iterative formula of the conjugate gradient method is given by (1.2), where α_k is a steplength which is computed by carrying out a line search, and d_k is the search direction defined by

$$d_{k+1} = \begin{cases} -g_k & \text{si } k = 1 \\ -g_{k+1} + \beta_k d_k & \text{si } k \geq 2 \end{cases} \tag{1.3}$$

where β_k is a scalar and $g(x)$ denotes $\nabla f(x)$. If f is a strictly convex quadratic function, namely,

$$f(x) = \frac{1}{2}x^T Hx + b^T x, \tag{1.3bis}$$

where H is a positive definite matrix and if α_k is the exact one-dimensional minimizer along the direction d_k, i.e.,

$$\alpha_k = \arg\min_{\alpha>0} \{f(x + \alpha d_k\} \tag{1.3tris}$$

then (1.2), (1.3), (1.3bis), (1.3tris) is called the linear conjugate gradient method. Otherwise, (1.2), (1.3) is called the nonlinear conjugate gradient method.

The line search in the non linear conjugate gradient methods is often based on the standard Wolfe conditions:

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

$$g_{k+1}^t d_k \ge \delta g_k^t d_k \tag{1.5}$$

where $0<\rho\le\delta<1$.

Conjugate gradient methods differ in their way of defining the scalar parameter β_k. In the literature, there have been proposed several choices for β_k which give rise to distinct conjugate gradient methods. The most well known conjugate gradient methods are the Hestenes–Stiefel (HS) method [14], the Fletcher–Reeves (FR) method [10], the Polak-Ribière-Polyak (PR) method [16], the Conjugate Descent method(CD) [10], the Liu-Storey (LS) method [15], the Dai-Yuan (DY) method [08], [09] and Hager and Zhang (HZ) method [13]. The update parameters of these methods are respectively specified as follows: $\beta_k^{HS}=\frac{g_{k+1}^T y_k}{d_k^T y_k}$, $\beta_k^{FR}=\frac{\|g_{k+1}\|^2}{\|g_k\|^2}$, $\beta_k^{PRP}=\frac{g_{k+1}^T y_k}{\|g_k\|^2}$, $\beta_k^{CD}=-\frac{\|g_{k+1}\|^2}{d_k^T g_k}$,

$\beta_k^{LS}=-\frac{g_{k+1}^T y_k}{d_k^T g_k}$, $\beta_k^{DY}=\frac{\|g_{k+1}\|^2}{d_k^T y_k}$, $\beta_k^{HZ}=\left(y_k-2d_k\frac{\|y_k\|^2}{d_k^T y_k}\right)^T\frac{g_{k+1}}{d_k^T y_k}$

Some of these methods, such as Fletcher and Reeves (FR) [10], Dai and Yuan (DY) [8] and Conjugate Descent (CD) [10] have strong convergence properties, but they may have modest practical performance due to jamming. On the other hand, the methods of Polak and Ribière and Polyak (PRP) [16], Hestenes and Stiefel (HS) [14] or Liu and Story (LS) [15] may not generally be convergent, but they often have better computational performance.

In the process of obtaining more robust and efficient conjugate gradient methods, some researchers suggested the hybrid conjugate gradient algorithm which combined the good features of the methods involve in the hybridization.

The first hybrid conjugate gradient method was given by Touati-Ahmed and Storey (1990) [20] to avoid jamming phenomenon.

The researchers were motived by the works of Andrei [3], [5]; Dai and Yuan [9] ; Zhang and Zhou [21]. Their parameter β_k^N is computed as a convex combination of β_k^{FR} and β_k^* other algorithms, i.e

$$\beta_k^N=(1-\theta_k)\beta_k^{FR}+\theta_k\beta_k^*$$

The Wolfe line search was employed to determine the step length $\alpha_k>0$ and the new method proved to be more robust numerical wise as compared to FR and other methods. The global convergence was establised under some suitable conditions.

In ([5]) Andrei has proposed a new hybrid conjugate gradient algorithm where the parameter β_k^A is computed as a convex combination of the Polak-Ribière- Polyak and the Dai- Yuan conjugate gradient algorithms i.e

$$\beta_k^A=(1-\theta_k)\beta_k^{PRP}+\theta_k\beta_k^{DY}$$

and θ_k is presented to satisfy the conjugacy condition

$$\theta_k=\theta_k^{CCOMB}=\frac{(y_k^t g_{k+1})(y_k^t s_k)-(y_k^t g_{k+1})(g_k^t g_k)}{(y_k^t g_{k+1})(y_k^t s_k)-\|g_{k+1}\|^2\|g_k\|^2}$$

where $s_k=x_{k+1}-x_k$. To satisfy Newton direction he takes

$$\theta_k=\theta_k^{NDOMB}=\frac{(y_k^t g_{k+1}-s_k^t g_{k+1})\|g_k\|^2-(y_k^t g_{k+1})(y_k^t s_k)}{\|g_{k+1}\|^2\|g_k\|^2-(y_k^t g_{k+1})(y_k^t s_k)}$$

but in the combination of HS and DY from Newton direction, he puts

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

On the other hand, from Newton direction with modified secant condition (Hybrid M-Andrei), Andrei has proposed another method

$$\beta_k^{HYBRIDM}=(1-\theta_k)\beta_k^{HS}+\theta_k\beta_k^{DY}$$

where

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

δ is parameter. In [17], [18] Salah Gazi Shareef and Hussein Ageel Khatab have introduced a new hybrid CG method

$$\beta_k^{New}=(1-\theta_k)\beta_k^{PRP}+\theta_k\beta_k^{BA}$$

where β_k^{BA} is selected in [5].

In this paper, we present another hybrid CG algorithm noted CGHLB (HLB is an abbreviation to Hadji; Laskri and Benzine), witch is a convex combination of the PRP ([16]) and RMIL+ ([17]) conjugate gradient algorithms.We are interested to combine these two methods in a hybrid CG algorithm because

PRP has good computational properties and RMIL+ has strong convergence properties. In section 2, we introduce our hybrid CG method and prove that it generates descent directions. In Section 3 we present and prove global convergence results. Numerical results and a conclusion are presented in section 4. By comparing numerically CGHLB with PRP and RMIL+ and by using the Dolan and More CPU performance, we deduce that CGHLB is more efficient.

2 Definitions of Function Spaces and Notation

3 A new hybrid conjugate gradient method

The iterates x_0, x_1, ...of our algorithm are computed by means of the recurrence (1.2) where the step size $\alpha_k > 0$ is determined according to the wolfe line search conditions (1.4) , (1.5). The directions d_k are generated by the rule:

$$d_k = \begin{cases} -g_0 & \text{if } k = 0 \\ -g_k + \beta_{K-1}^{HLB} d_{k-1} & \text{if } k \geq 1 \end{cases} \quad (2.1)$$

where

$$\beta_k^{HLB} = (1-\theta_k)\,\beta_k^{PRP} + \theta_k \beta_k^{RMIL+}$$

i.e

$$\beta_k^{HLB} = (1-\theta_k)\,\frac{g_{k+1}^t y_k}{\|g_k\|^2} + \theta_k \frac{g_{k+1}^t\,(g_{k+1}-g_k-d_k)}{\|d_k\|^2} \quad (2.2)$$

HLB is an abbreviation to Hadji; Laskri and Benzine; θ_k is a scalar parameter which will be determined in a specific way to be described in the folloing section. Observe that if $\theta_k = 0$ then $\beta_k^{HLB} = \beta_k^{PRP}$ and if $\theta_k = 1$, then $\beta_k^{HLB} = \beta_k^{RMIL+}$. On the other hand if $0 < \theta_k < 1$, then β_k^{HLB} is a convex combination of $\beta_k^{PRP}$and β_k^{RMIL+}. The parameter θ_k is selected in such away that at every iteration the conjugacy condition is satisfied . It can be noted that,

$$d_{k+1} = -g_{k+1} + (1-\theta_k)\,\frac{g_{k+1}^t y_k}{\|g_k\|^2} d_k + \theta_k \frac{g_{k+1}^t\,(g_{k+1}-g_k-d_k)}{\|d_k\|^2} d_k \quad (2.3)$$

so multiply both sides of above equation by y_k and by using the conjugacy condition $\left(d_{k+1}^t y_k = 0\right)$ we have:

$$0 = -g_{k+1}^t y_k + (1-\theta_k)\,\frac{g_{k+1}^t y_k}{\|g_k\|^2} d_k^t y_k + \theta_k \frac{g_{k+1}^t\,(g_{k+1}-g_k-d_k)}{\|d_k\|^2} d_k^t \quad (2,4)$$

After a simple calculation we get

$$\theta_k = \frac{g_{k+1}^t g_k \|g_k\|^2 \|d_k\|^2 - \left(g_{k+1}^t y_k\right)\left(d_k^t y_k\right)\|d_k\|^2}{\left(g_{k+1}^t y_k - g_{k+1}^t d_k\right)\|g_k\|^2 - \left(g_{k+1}^t y_k\right)\left(d_k^t y_k\right)\|d_k\|^2} \quad (2.5)$$

So, to ensure the convergence of this method when the parameter θ_k goes out of interval $]0,1[$; i.e. when $\theta_k \leq 0$ or $\theta_k \geq 1$, we prefer to take β_k^{HLB} as following:

$$\beta_k^{HLB} = \begin{cases} (1-\theta_k)\,\beta_k^{PRP} + \theta_k \beta_k^{RMIL+} & \text{if } 0 < \theta_k < 1 \\ \beta_k^{PRP} & \text{if } \theta_k \leq 0 \\ \beta_k^{RMIL+} & \text{if } \theta_k \geq 1 \end{cases} \quad (2.5(\text{bis}))$$

We are now able to present our new algorithm, the Conjugate Gradient CGHLB Algorithm: **CGHLB Algorithm**

Step1: set, $k = 0$, select the initial point $x_o \in \mathbb{R}^n$.select the parameters $0 < \rho \leq \delta < 1$, and $\varepsilon > 0$

compute $f(x_0)$, and $g_0 = \nabla f(x_0)$. consider $d_0 = -g_0$

Step2: Test for continuation of iterations:

If $\|g_k\| \leq \varepsilon$ then stop else set . $d_k = -g_k$ **Step3: Line search**:

Compute $\alpha_k > 0$ satisfying the Wolfe line search condition $(1,4)$ and $(1,5)$ and update the variables, $x_{k+1} = x_k + \alpha_k d_k$; compute $f(x_{k+1})$, g_{k+1} and $s_k = x_{k+1} - x_k$; $y_k = g_{k+1} - g_k$. **Step4: θ_k Parameter computation**:

If $\left(g_{k+1}^t y_k - g_{k+1}^t d_k\right)\|g_k\|^2 - \left(g_{k+1}^t y_k\right)\left(d_k^t y_k\right)\|d_k\|^2 = 0$; then set $\theta_k = 0$, otherwise, compute θ_k as in (2.5).

Step5:β_k^{HLB} conjugate gradient parameter computation:

If $0 < \theta_k < 1$, then compute β_k^{HLB} as in (2.2).

If $\theta_k \geq 1$, then set $\beta_k^{HLB} = \beta_k^{RMIL+}$

If $\theta_k \leq 0$then set $\beta_k^{HLB} = \beta_k^{PRP}$

Step6:Direction computation:

compute $d_{k+1} = -g_{k+1} + \beta_k^{HLB} d_k$

Set k=k+1 and go to step 3.

The following theorem shows that our method assures the descent condition, when $0 < \theta_k < 1$

Theorem 1 *In the algorithm* (1.2) , (1.3) *and* (2.5) *assume that* d_k *is a descent direction* $\left(g_k^t d_k < 0\right)$, *and* α_k *is determined by the Wolfe line search* (1.4) ; (1.5). *If* $0 < \theta_k < 1$ *then the direction* d_{k+1}*given by* (2.3) *is a descent direction.*

Proof 2 *Multiply both sides of* $(2,3)$ *by* g_{k+1} *we have:*

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

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

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

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

since $0 < \theta_k < 1$ *then*

$$g_{k+1}^T d_{k+1} \le \left[-\|g_{k+1}\|^2 + \frac{g_{k+1}^t y_k}{\|g_k\|^2} d_k^t g_{k+1}\right] + \left[-\|g_{k+1}\|^2 + \frac{g_{k+1}^t (g_{k+1} - g_k - d_k)}{\|d_k\|^2} d_k^t g_{k+1}\right] \quad (2.6)$$

If the step length α_k *is chosen by an exact line search. Then* $g_{k+1}^T d_k = 0$.

If the step length α_k *is chosen by an inexact line search* $\left(g_{k+1}^T d_k \neq 0\right)$ *then we have:*

$$g_{k+1}^T d_{k+1} < 0$$

because the algorithms of (PRP) *and* $(RMIL+)$ *satisfied the descent property.*

The proof is completed.

4 Global convergence properties

The following assumptions are often needed to prove the convergence of the nonlinear CG

Assumption 1

(i) The level set $\Omega = \{x \in \mathbb{R}^n / f(x) \le f(x_0)\}$*is bounded, where* x_0 *is the starting point.*

(ii) In some neighborhood N *of* Ω, *the objective function is continuously differentiable and its gradient is Lipschitz continuous, namely, there exists a constant* $l > 0$ *such that:*

$$\|g(x) - g(y)\| \le l\|x - y\| \quad for\ any\ x, y \in N$$

Under these assumptions on f , there exists a constant μ such that $\|g(x)\| \le \mu$, for all $x \in \Omega$.

Lemma 3 [23] *Suppose Assumption 1 holds, 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. If*

$$\sum_{k=1}^{\infty} \frac{1}{\|d_k\|^2} = +\infty \quad (3.1)$$

then

$$\liminf_{k\to\infty} \|g_k\| = 0 \quad (3.2)$$

Assume that the function f is uniformly convex function , i.e, there exists a constant $\Gamma \ge 0$ such that,

$$for\ all\ x, y \in \Omega : (\nabla f(x) - \nabla f(y))^t (x - y) \ge \Gamma \|x - y\|^2 \quad (3.3)$$

and the steplength α_k is given by the strong Wolfe line search.

$$f(x_k + \alpha_k d_k) - f(x_k) \le \sigma_1 \alpha_k g_k^t d_k \quad (3.4)$$

$$\left|g_{k+1}^t d_k\right| \le -\sigma_2 g_k^t d_k \quad (3.5)$$

For uniformly convex function which satisfy the above assumptions, we can prove that the norm of d_{k+1} given by (2.3) is bounded above.

Using the above lemma, we obtain the following theorem.

Theorem 4 *Suppose that Assumption 1 holds. Consider the algorithm* $(1.2)\,;(2.3)$*and* (2.5) *,where* $0 \le \theta_k \le 1$ *and* α_k *is obtained by the strong Wolfe line search.*(3.4) *and* (3.5) .

If d_k *tends to zero and there exists non negative constants* η_1 *and* η_2 *such that;*

$$\|g_k\|^2 \ge \eta_1 \|s_k\|^2 \ and\ \|g_{k+1}\|^2 \le \eta_2 \|s_k\| \quad (3.6)$$

and f *is uniformly convex function, then*

$$\lim_{k\to\infty} g_k = 0 \quad (3.7)$$

Proof 5 *From* $(3,3)$ *it follows that*

$$y_k^t s_k \ge \Gamma \|s_k\|^2$$

since $0 \leq \theta_k \leq 1$ *, from uniform convexity and* (3.6) *we have*

$$\left|\beta_k^{HLB}\right| \leq \left|\frac{g_{k+1}^t y_k}{\|g_k\|^2}\right| + \left|\frac{g_{k+1}^t (g_{k+1} - g_k - d_k)}{\|d_k\|^2}\right|$$

$$\leq \frac{\left|g_{k+1}^t y_k\right|}{\|g_k\|^2} + \frac{\left|g_{k+1}^t y_k\right|}{\|d_k\|^2} + \frac{\left|g_{k+1}^t d_k\right|}{\|d_k\|^2}$$

$$\leq \frac{\|g_{k+1}\| \|y_k\|}{\|g_k\|^2} + \frac{\|g_{k+1}\| \|y_k\|}{\|d_k\|^2} + \frac{\|g_{k+1}\| \|d_k\|}{\|d_k\|^2}$$

from Lipschitz condition

$$\|y_k\| \leq l \|s_k\|$$

$$\left|\beta_k^{HLB}\right| \leq \frac{\|g_{k+1}\| \|y_k\|}{\eta_1 \|s_k\|^2} + \frac{\|g_{k+1}\| \|y_k\|}{\|d_k\|^2} + \frac{\|g_{k+1}\|}{\|d_k\|}$$

$$\leq \frac{\mu l \|s_k\|}{\eta_1 \|s_k\|^2} + \frac{\mu l \|s_k\| \alpha_k^2}{\|s_k\|^2} + \frac{\mu \alpha_k}{\|s_k\|}$$

$$= \frac{\mu l}{\eta_1 \|s_k\|} + \frac{\mu l \alpha_k^2}{\|s_k\|} + \frac{\mu \alpha_k}{\|s_k\|}$$

Hence

$$\|d_{k+1}\| \leq \|g_{k+1}\| + \left|\beta_k^{HLB}\right| \|d_k\|$$

$$\leq \mu + \frac{\mu l \|s_k\|}{\eta_1 \alpha_k \|s_k\|} + \frac{\mu l \|s_k\| \alpha_k^2}{\alpha_k \|s_k\|} + \frac{\mu \alpha_k \|s_k\|}{\alpha_k \|s_k\|}$$

$$= 2\mu + \mu l \alpha_k + \frac{\mu l}{\eta_1 \alpha_k}$$

which implies that (3.1) *is true.Therefore, by lemma 1 we have* (3.2)*, which for uniformly convex functions is equivalent to* (3.7) .

5 Numerical results and discussions

In this section we report some numerical results obtained with a MATLAB implementation of conjugate gradient algorithms and their new variants. All codes are written in Matlab on a Workstation Intel Pentium 4 with 1.8 GHz. We selected a number of 75 large-scale unconstrained optimization test functions in generalized or extended form [6] (some from CUTE library [8]). For each test function we have considered ten numerical experiments with the number of variables $n = 1000, 2000, ..., 10000$. In the following we present the numerical performance of CG codes corresponding to different formula for β_k computation. All algorithms implement the Wolfe line search conditions with $\rho = 0.0001$ and $\sigma = 0.9$, and the same stopping criterion $\|g_k\|_\infty \leq 10^{-6}$, 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 $i = 1, ..., 750$, respectively. We say that, in the particular problem i, the performance of $ALG1$ was better than the performance of $ALG2$ if: $\left|f_i^{ALG1} - f_i^{ALG2}\right| < 10^{-6}$, 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.

For each algorithm, we plot the fraction of problems for which the algorithm is within a factor s of the best cpu time. Relative to performance profiles, the top curve corresponds to the method that solved the most problems in a time that was within a factor τ of the best time. By comparing numerically CGHLB with PRP and RMIL+ (see Fig. 1 and fig2) and by using the Dolan and More CPU performance, we deduce that CGHLB is more efficient.

References:

[1] Al-Baali, ”Descent property and global convergence of Fletcher-Reeves method with inexact line serch”, IMA.J.Numer. Anal., 5;121-124.(1985).

[2] A.Y. Al-Bayati, and N.H. Al-Assady, ”Conjugate Gradient Method”, Technical Research report, No. 1,school of computer studies , Leeds Uuiversity(1986).

[3] N. Andrei, ”Hybrid Conjugate Gradient Algorithm for Unconstrained Optimization”, J.Optim. Theory Appl., 141, , 249-264.(2009).

[4] N. Andrei, ”An unconstrained optimization test functions collection”, Advanced Modeling and Optimization, An ElectronicE. Polak, G. Ribiere, ”Note sur la convergence des méthodes de directions conjugées”, *Rev. Francaise informatique et recherche operationnelle, vol 16, pp. 35-43,*.(1969)

[5] N. Andrei, ”Another hybrid conjugate gradient algorithm for unconstrained optimization”, Numer. Algorithms 47, 143–156 (2008).

[6] I. Bongartz, A.R. Conn, N.I.M. Gould, P.L. Toint, ”CUTE: constrained and unconstrained testing environments”, ACM Transactions on Mathematical Software 21 (1995) 123–160.

[7] J.W. Daniel, ”The conjugate gradient method for linear and nonlinear operator equations”, SIAM J. Numer. Anal. 4 (1967) 18.10–26.

[8] Y.H. Dai and Y. Yuan, “A nonlinear conjugate gradient method with a strong global convergence property”, SIAM J. Optimization, 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] R. Fletcher, ”Practical Methods of Optimization,” vol. 1: Unconstrained Optimization, John Wiley & Sons, New York, 1987.

[11] S. Gazi and H. Khatab, ”New iterative conjugate gradient method for nonlinear unconstrained optimization using homptopy technique”, IOSR journal of Mathmatics , pp 78-82. (2014).

[12] W.W. Hager and H. Zhang ”A survey of nonlinear conjugate gradient methods”, Pacific J.Optim., submitted(2005).

[13] W. W. Hager and H. Zhang, ”A new conjugate gradient method with guaranteed descent and an effcient line search”, November 17, 2003 (to appear in SIAM J. Optim.)

[14] M. Hestenes, ”Methods of conjugate gradients for solving linear systems”, Research Jr. of the National Bureau of Standards 49 22,(1952) 409–436.

[15] Y. Liu, C. Storey, ”Efficient generalized conjugate gradient algorithms”, Part 1: Theory. J. Optim. Theory Appl. 69, 129–137 (1991)

[16] B.T. Polyak, ”The conjugate gradient method in extremal problems”, USSR Comput. Math. Math. Phys. 9 (1969) 94–112.

[17] M. Rivaie, M. Mustafa, A. Abashar, ”A new class of nonlinear conjugate coeffients with exact and inexact line searchs”, Appl.Math.com 1152-1163(2015).

[18] M. Rivaie, M. Mustafa, L.W. June, I. Mohd, ”A new class of nonlinear conjugate gradient coefficient with global convergence properties”, Appl. Math. Comp.218 11323–11332.(2012) International Journal 10,(2008) 147–161.

[19] D.F. Shanno, ”Conjugate gradient methods with inexact searches”, Math. Oper. Res. 3 (1978) 244–256.

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

[21] L. Zhang and W. Zhou, ”Two descent hybrid conjugate gradient methodq for optimization”, Journal of computational and Applied Mathematics. 216(1), 251-264 (2008).

[22] D. Zhifeng, ”Comments on hybrid conjugate gradient algorithm for unconstrained optimization” J Optim Theory App.(2017) 175: 286-291.

[23] G. Zoutendijk, ”Nonlinear Programming”, Computational Methods, in: Integer and Nonlinear Programming (J. Abadie, ed.),North-Holland, Amsterdam, , pp. 37-86. (1970).

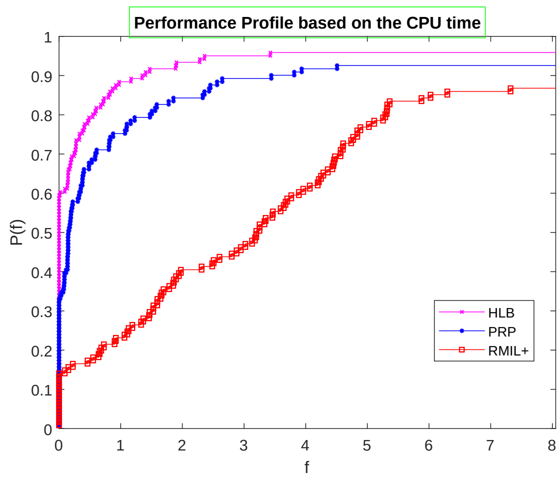

Figure 1: Performance profile based on the CPU time

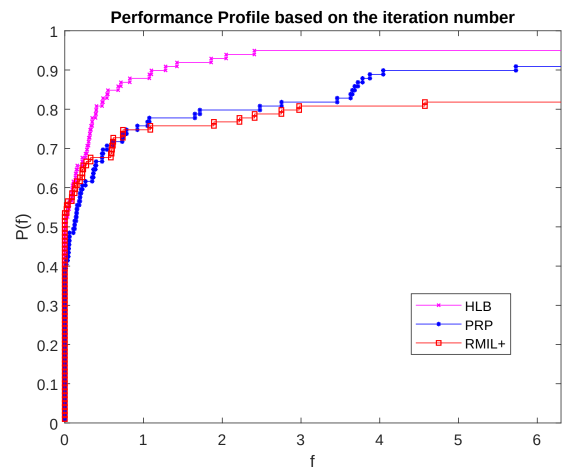

Figure 2: Performance profile based on the iteration number