BEHAVIOR OF THE COMBINATION OF PRP AND HZ METHODS FOR UNCONSTRAINED OPTIMIZATION

SARRA DELLADJI*
MOHAMMED BELLOUFI AND BADREDDINE SELAMI
Laboratory Informatics and Mathematics (LiM)
Mohamed Cherif Messaadia University
Souk Ahras, 41000, Algeria

(Communicated by Alexander Gornov)

ABSTRACT. To achieve a conjugate gradient method which is strong in theory and efficient in practice for solving unconstrained optimization problem, we propose a hybridization of the Hager and Zhang (HZ) and Polak-Ribièere and Polyak (PRP) conjugate gradient methods which possesses an important property of the well known PRP method: the tendency to turn towards the steepest descent direction if a small step is generated away from the solution, averting a sequence of tiny steps from happening, the new scalar $\beta_k$ is obtained by convex combination of PRP and HZ under the Wolfe line search we prove the sufficient descent and the global convergence. Numerical results are reported to show the effectiveness of our procedure.

1. Introduction. Let us consider the nonlinear unconstrained optimization problem:

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

where $f : \mathbb{R}^n \to \mathbb{R}$ is continuously differentiable function and its gradient $g(x) = \nabla f(x)$ is available. The nonlinear conjugate gradient (CG) method is highly useful for solving this kind of problems because of its simplicity and its very low memory requirement [4]. The iterative formula of the CG methods is given by:

$$x_{k+1} = x_k + s_k, \quad s_k = \alpha_k d_k, \quad k = 0, 1, \ldots, n.$$  

where $x_k$ is the $k^{th}$ iterate point, $\alpha_k$ is step length which is obtained by carrying out some linear search, such as exact or inexact line search. In practical computation, exact line search is consumption time and the workload is very large, so we usually take the following inexact line search ([16], [17]). Usually, a major inexact line search is the strong Wolfe line search. The strong Wolfe line search is to find the step-length $\alpha_k$ in (2) satisfying:

$$f(x_k + \alpha_k d_k) - f(x_k) \leq \delta \alpha_k g_k^T d_k,$$

$$|g(x_k + \alpha_k d_k)^T d_k| \leq \sigma |g_k^T d_k|,$$

where parameters $\delta$ and $\sigma$ satisfy $0 < \delta < \sigma < 1$.
And $d_k$ is the search direction generated by the rule:

$$d_k = \begin{cases} -g_k & \text{for } k = 0 \\ -g_k + \beta_{k-1}d_{k-1} & \text{for } k \geq 1, \end{cases}$$

where $g_k := \nabla f(x_k)$ is the gradient of $f$ at $x_k$, and $\beta_k \in \mathbb{R}$ is a parameter which determines the different CGMs. One of the efficient methods which has possess an approximate restart feature when jamming occurs has proposed by Polak, Ribière and Polyak ([13], [14]) (PRP) with the following CG parameter:

$$\beta_{PRP} = \frac{g_{k+1}^T y_k}{||y_k||^2},$$

where $||.||$ stands for the Euclidean norm, $y_k = g_{k+1} - g_k$. In spite of the numerical efficiency of the PRP method, Powell [15] constructed a counter example demonstrating the method can cycle infinitely. One of the conjugate gradient method which is strong in theory is suggested by Hager and Zhan [10] with the following formula of $\beta_k$:

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

(7)

To achieve a method which is posses a good performance and strong convergence we suggest a hybridization of PRP and HZ methods as a convex combination to exploit the interesting features of each method.

Under the strong Wolfe line search with the parameter $\sigma \leq \frac{1}{2}$, Al-Baali [1] proved that the FR method satisfies the sufficient descent condition and converges globally for general objective functions. Dai and Yuan [7] shown that the DY method is descent and globally convergent if the Wolfe line search is used. In contrary, the PRP method and the HS (Hestenes and Stiefel) method are generally regarded to be two of the most efficient conjugate gradient methods in practical computation, but their convergence properties are not so good.

Recently, Andrei [2] introduced a new hybrid conjugate gradient method (denoted as HYBRID method) based on HS and DY methods for large-scaled unconstrained optimization problems. In [12], Liu and al. discussed the global convergence of the LS (Liu and Storey) and DY with inexact line search for nonconvex unconstrained optimization. Snezana S. Djordjevic [8] analyzed the global convergence of a convex combination of FR (Fletcher and Reeves) and PRP methods with sufficient descent property.

The paper is organized as follows, in section 2 we obtain the parameter $\theta_k$, discuss the sufficient descent property and give our specific algorithm of the proposed method. In Section 3, the global convergence of the proposed method is established. Preliminary numerical results are presented in Section 4. Finally, we make conclusions.

2. A hybridization of the PRP and HZ methods. In this section, we deal with the following convex combination of the CG parameters of the HZ and PRP methods:

$$\beta_k^{hPRPHZ} = (1 - \theta_k)\beta_k^{HZ} + \theta_k\beta_k^{PRP},$$

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

(8)

in which $\theta_k \in [0, 1]$ is called the hybridization parameter. Note that if $\theta_k = 0$ then $\beta_k^{hPRPHZ} = \beta_k^{HZ}$, and if $\theta_k = 1$, then $\beta_k^{hPRPHZ} = \beta_k^{PRP}$. On the other hand if
0 < \theta_k < 1 \) then the parameter \( \theta_k \) is selected in such a way that at every iteration the conjugacy condition \( (d_{k+1}^T y_k = 0) \) is satisfied independently of the line search. Clearly

\[
d_{k+1} = -g_{k+1} + (1 - \theta_k) \frac{1}{d_k^T y_k} (y_k^T g_{k+1} - 2 \frac{\|y_k\|^2}{d_k^T y_k} d_k^T g_{k+1}) d_k + \theta_k \frac{g_{k+1}^T y_k}{\|y_k\|^2} d_k,
\]

(9)
multiply both sides of above equation by \( y_k \), implies

\[
0 = -g_{k+1}^T y_k + (1 - \theta_k)(y_k^T g_{k+1} - 2 \frac{\|y_k\|^2}{d_k^T y_k} d_k^T g_{k+1}) + \theta_k \frac{g_{k+1}^T y_k}{\|y_k\|^2} d_k^T y_k,
\]

after some algebra we have:

\[
\theta_k = \frac{2 \frac{\|y_k\|^2}{d_k^T y_k} d_k^T y_{k+1}}{g_{k+1}^T y_k - y_k^T g_{k+1} + 2 \frac{\|y_k\|^2}{d_k^T y_k} d_k^T g_{k+1}}.
\]

(10)

It possible that \( \theta_k \), calculated as in (10) has the values outside the interval \([0,1]\). So we fixe it:

\[
\theta_k = \begin{cases} 
0 & \text{if } \frac{2 \frac{\|y_k\|^2}{d_k^T y_k} d_k^T y_{k+1}}{g_{k+1}^T y_k - y_k^T g_{k+1} + 2 \frac{\|y_k\|^2}{d_k^T y_k} d_k^T g_{k+1}} \leq 0, \\
1 & \text{if } \frac{2 \frac{\|y_k\|^2}{d_k^T y_k} d_k^T y_{k+1}}{g_{k+1}^T y_k - y_k^T g_{k+1} + 2 \frac{\|y_k\|^2}{d_k^T y_k} d_k^T g_{k+1}} \geq 1, \\
\frac{y_{k+1}^T y_k - y_k^T g_{k+1} + 2 \frac{\|y_k\|^2}{d_k^T y_k} d_k^T g_{k+1}}{\|y_k\|^2 d_k^T y_k} & \text{else}.
\end{cases}
\]

(11)

**Theorem 2.1.** \([8]\) If the relations (8) and (9) hold, then

\[
d_{k+1}^{PRPHZ} = (1 - \theta_k) d_{k+1}^{HZ} + \theta_k d_{k+1}^{PRP}.
\]

(12)

**Proof.** We have \( d_{k+1}^{PRPHZ} = -g_{k+1} + \theta_k^{PRPHZ} d_k \). After adding and subtracting \((\theta_k g_{k+1})\) we obtain

\[
d_{k+1}^{PRPHZ} = (1 - \theta_k)(-g_{k+1} + \beta_k^{PRPHZ} d_k) + \theta_k(-g_{k+1} + \beta_k^{PRP} d_k),
\]

(13)

implies

\[
d_{k+1}^{PRPHZ} = (1 - \theta_k) d_{k+1}^{HZ} + \theta_k d_{k+1}^{PRP}.
\]

(14)

\[\square\]

**Assumption 1.** The level set \( S = \{x \in \mathbb{R}^n \mid f(x) \leq f(x_0)\} \) is bounded, i.e. there exists a constant \( B > 0 \), such that

\[
\|x\| \leq B, \text{ for all } x \in S.
\]

(15)

**Assumption 2.** In a neighborhood \( N \) of \( S \) the function \( f \) is continuously differentiable and its gradient \( \nabla f(x) \) is Lipschitz continuous, i.e. there exists a constant \( 0 < L < \infty \) such that

\[
\|\nabla f(x) - \nabla f(y)\| \leq L \|x - y\| \text{ for all } x, y \in N.
\]

(16)
Under these assumptions, there exists a constant $\Gamma \geq 0$, such that
\[ \|\nabla f(x)\| \leq \Gamma, \quad (17) \]
for all $x \in S$ [3].

The next subsection prove the sufficient descent of our hybridation:

2.1. **Sufficient descent condition.** According to the theorem (2.1) we have:

- Firstly, if $\theta_k = 0$ then $d_k^{P R P H Z} = d_k^{H Z}$, the sufficient descent condition holds for the hybrid method, if it holds for HZ method. William W. Hager and Hongchao Zhang prove in [10] that $d_k^{H Z}$ satisfies the sufficient descent condition for all $k$, and the details as follows:

**Theorem 2.2.** [10] If $d_k^{T}y_k \neq 0$, and
\[ d_{k+1} = -g_{k+1} + \tau d_k, \quad d_0 = -g_0 \quad \forall \tau \in [\beta_k^{H Z}, \max \{ \beta_k^{H Z}, 0 \}], \quad (18) \]
then
\[ g_{k+1}^{T}d_{k+1}^{H Z} \leq -\frac{7}{8}\|g_{k+1}\|^2. \quad (19) \]

**Proof.** According to (18) we have two case:
The first one if $\beta_k^{H Z} > 0$ then $\tau = \beta_k^{H Z}$ and
\[ g_{k+1}^{T}d_{k+1} = -\|g_{k+1}\|^2 + \beta_k^{H Z}T g_{k+1}d_k \]
\[ = -\|g_{k+1}\|^2 + \left( \frac{g_k^{T}g_{k+1}}{d_k^{T}y_k} - \frac{2\|y_k\|^2d_k^{T}g_{k+1}}{(d_k^{T}y_k)^2} \right) g_{k+1}^{T}d_k \]
\[ = -\|g_{k+1}\|^2(d_k^{T}y_k)^2 + \left( \frac{g_k^{T}g_{k+1}(g_{k+1}^{T}d_k)(d_k^{T}y_k) - 2\|y_k\|^2(d_k^{T}g_{k+1})^2}{(d_k^{T}y_k)^2} \right) \]
\[ \leq \frac{1}{2}(d_k^{T}y_k)^2(-\|g_{k+1}\|^2(\frac{d_k^{T}y_k)^2}{2} + \frac{1}{8}(d_k^{T}y_k)^2\|g_{k+1}\|^2 + 2\|g_{k+1}d_k\|^2) \]
\[ \leq -\frac{7}{8}\|g_{k+1}\|^2. \quad (20) \]

The second case if $\beta_k^{H Z} \leq 0$ then $\tau \in [\beta_k^{H Z}, 0]$ and
\[ g_{k+1}^{T}d_{k+1} = -\|g_{k+1}\|^2 + \tau g_{k+1}^{T}d_k, \]
if $g_{k+1}^{T}d_k < 0$ then from (20) and (21) we obtain:
\[ g_{k+1}^{T}d_{k+1} \leq -\|g_{k+1}\|^2 + \beta_k^{H Z}g_{k+1}^{T}d_k \]
\[ \leq -\frac{7}{8}\|g_{k+1}\|^2. \quad (22) \]

Else the aim follows immediately because $\tau < 0$.

- Secondly, if $\theta_k = 1$ then $d_k^{P R P H Z} = d_k^{P R P}$.

So, if the sufficient descent holds for PRP method, it holds for hPRPHZ method. The following theorem [8] prove the sufficient descent for PRP method.
Step 7: Compute the initial guess, Condition

Step 4: If Generate the next iterate by

Step 3: Compute

Step 2: Set $d_0 = -g_0$, the initial guess $\alpha_0 = \frac{1}{||g_0||}$ and $k = 0$.

Step 1: If $||g_k|| < \epsilon$ then Stop, else go to Step 2.

Step 2: Compute $\alpha_k$ by the strong Wolfe line search (3), (4).

Step 3: Generate the next iterate by $x_{k+1} = x_k + \alpha_k g_k$.

Step 4: Compute $g_{k+1} = \nabla f(x_{k+1})$ and $y_k = g_{k+1} - g_k$.

Step 5: Compute $\beta_k$ as in (8).

Step 6: Compute $d = -g_k + \beta_k d_{PRP_{k+1}}$. If the restart criterion of Powell condition

\[ |g_k^Tg_k| \geq 0.2|g_k||^2, \]  

is satisfied, then $d_{k+1} = -g_{k+1}$, else define $d_{k+1} = d$.

Step 7: Compute the initial guess $\alpha_k = \alpha_{k-1} \frac{||d_{k-1}||}{||d_k||}$.

Step 8: Put $k = k + 1$ and go to Step 1.

Theorem 2.3. [6] Assume that (15), (16) hold, let $\eta$ a non negative constant such that:

\[ ||g_k||^2 \geq \eta ||s_k||^2, \eta \geq L. \]  

Then $d_{k+1}^{PRP}$ satisfies the sufficient descent condition for all $k$.

Proof. We have by using Cauchy-Bunyakovsky-Schwartz inequality:

\[ g_k^Tg_{k+1} = \frac{1}{2} ||g_{k+1}||^2 + \frac{1}{2} ||g_k||^2 + \frac{1}{2} \beta_k ||g_{k+1}|| ||g_{k+1}|| \]

From (16) we have $y_k \leq L ||s_k||$, so:

\[ g_k^Tg_{k+1} \leq -||g_{k+1}||^2 + ||g_{k+1}||^2 L ||s_k||^2 \]

by (23):

\[ g_k^Tg_{k+1} \leq -(1 - \frac{L}{\eta}) ||g_{k+1}||^2. \]

Finally, for $0 < \theta_k < 1$ there exist $\lambda_1, \lambda_2$ in which that $0 < \lambda_1 \leq \theta_k \leq \lambda_2 < 1$, we get:

\[ g_k^Tg_{k+1} \leq \eta g_k^Tg_{k+1} + (1 - \eta_2) g_{k+1}^Tg_{k+1}, \]

We evidently can achieve that there exists a number $k > 0$, such that

\[ g_k^Tg_{k+1} \leq -k ||g_{k+1}||^2. \]  

2.2. Algorithm (hPRPHZ). Initialization: Choose an initial point $x_0 \in \mathbb{R}^n$, $\epsilon > 0$. Compute $f(x_0)$ and $g_0 = \nabla f(x_0)$.

Step 1: If $||g_k|| < \epsilon$ then Stop, else go to Step 2.

Step 2: Compute $\alpha_k$ by the strong Wolfe line search (3), (4).

Step 3: Generate the next iterate by $x_{k+1} = x_k + \alpha_k g_k$.

Step 4: Compute $g_{k+1} = \nabla f(x_{k+1})$ and $y_k = g_{k+1} - g_k$.

Step 5: Compute $\beta_k$ as in (8).

Step 6: Compute $d = -g_k + \beta_k d_{PRP_{k+1}}$. If the restart criterion of Powell condition

\[ |g_k^Tg_k| \geq 0.2|g_k||^2, \]  

is satisfied, then $d_{k+1} = -g_{k+1}$, else define $d_{k+1} = d$.

Step 7: Compute the initial guess $\alpha_k = \alpha_{k-1} \frac{||d_{k-1}||}{||d_k||}$.

Step 8: Put $k = k + 1$ and go to Step 1.
3. Global convergence. The following lemma gives the Zoutendijk condition [18], and a detailed proof can be found in [11].

**Lemma 3.1.** [12] Suppose that Assumption 1, Assumption 2 holds. If $d_k$ is a descent direction and the step size $\alpha_k$ satisfies

$$ g_{k+1}^T d_k \geq \sigma g_k^T d_k, \sigma < 1, \tag{29} $$

then

$$ \alpha_k \geq \frac{1 - \sigma}{L} \frac{|d_k^T g_k|}{||d_k||^2}. \tag{30} $$

**Proof.** Through (29), the Cauchy-Bunyakovsky-Schwartz inequality and (16), it holds that

$$ -(1 - \sigma) g_k^T d_k \leq d_k^T (g_{k+1} - g_k) \leq L\alpha_k ||d_k||^2. $$

Since $d_k$ is a descent direction and $\sigma < 1$, then the assertion (30) holds.

Obviously, from the strong Wolfe condition and (27), the step length $\alpha_k$ satisfies (30). According to the assumptions (1) and (2) and (27), it is easy to obtain that $g_k^T d_k \neq 0$ for all $k \geq 0$. Thus, $\alpha_k = 0$ does not satisfy (4). This indicates that $\alpha_k = 0$ obtained in the hPRPHZ method is not equal to zero, i.e., there exists a constant $\lambda > 0$ such that

$$ \alpha_k \geq \lambda, \forall k \geq 0. \tag{31} $$

**Lemma 3.2.** Suppose that Assumptions (1) and (2) holds. Consider common iterate (2), where $d_k$ is a descent direction and $\alpha_k$ satisfies the Wolfe line search (3). Then the zoutendijk condition

$$ \sum_{k=1}^{\infty} \frac{(g_k^T d_k)^2}{||d_k||^2} < \infty, \tag{32} $$

holds.

The following theorem gives the global convergence of hPRPHZ method.

**Theorem 3.3.** Suppose that Assumption (1) and (2) hold. Let $\{x_k\}$ be generated by Algorithm hPRPHZ. Then

$$ \lim_{k \to \infty} \inf ||g_k|| = 0. \tag{33} $$

**Proof.** Suppose by contradiction that (33) is false. Then there exists a constant $c > 0$ in which

$$ ||g_k||^2 \geq c, \forall k \text{ sufficiently large.} \tag{34} $$

According to (16), we get

$$ ||y_k|| = ||g_{k+1} - g_k|| \leq LD, \tag{35} $$

where $D = \max \{||x - y||, x, y \in N\}$ is the diameter of $N$. By using (4) and (27), we have

$$ d_k^T y_k \geq \sigma d_k^T g_k - d_k^T g_k \geq (1 - \sigma)k ||g_k||^2 \geq (1 - \sigma)kc. \tag{36} $$
From (9), (17), (34) (35) and (36) we obtain

\[ |\beta^H_k| \leq |\beta^H_k| + |\beta^P_k| \]

\[ \leq \frac{1}{d_k y_k} (||y_k||g_k||g_{k+1}|| + 2||g_k||d_k ||g_k||g_{k+1}||) + \frac{||g_{k+1}||g_k||}{||g_k||^2}, \]

\[ \leq \frac{LDT}{(1-\sigma)kc} (1 + 2LD^2/\lambda(1-\sigma)kc) + \frac{\Gamma LD}{c} = Q. \]

Also, from (8) and (31), we have

\[ ||d_{k+1}|| \leq ||g_{k+1}|| + |\beta^H_k|g_k||d_k|| = ||g_{k+1}|| + \frac{|\beta^H_k|g_k||d_k||}{\alpha_k} \]

\[ \leq \Gamma + \frac{QD}{\lambda} = W. \]

So:

\[ ||d_{k+1}|| \leq W \Rightarrow \sum_{k \geq 0} \frac{1}{||d_k||^2} = +\infty \]

\[ \Rightarrow \sum_{k \geq 0} \frac{(d_k^T g_k)^2}{||d_k||^2} = +\infty. \]

Which contradicts Lemma (3.2), therefore the claim (33) is proved.

4. Numerical results. In this section, we report some numerical experiments. We test the PRP and HZ methods on problems in the CUTE [5] library and compare their performance to that of the hPRPHZ method. For the numerical tests, the parameters in the strong Wolfe line searches are chosen to be \( \sigma = 0.9; \delta = 0.0001. \) We stop the iteration if the inequality \( \|g(x_k)\| \leq 10^{-6} \) is satisfied. In this paper, all codes were written in MATLAB and run on PC with Intel(r) Core(tm) i7-2670QM CPU @ 2.20GHz 2.20GHz processor and 4GB RAM memory and windows 10 Pr system.

Convenient for comparison, all tests are done under a variant of generalized Wolfe line Search as follows:

Table 1 list numerical results. The meaning of each column is as follows:

- “problem” the name of the test problem
- “n” the dimension of the test problem
- “iter” the number of iterations
- “time” the CPU time in seconds.

Figs. 1 and 2 show the performance of these methods relative to Iter and time (CPU time), which were evaluated using the profiles of Dolan and Moré [9]. Benchmark results are generated by running a solver on a set \( P \) of problems and recording information of interest Iter and Tcpu. Let \( S \) be the set of solvers in comparison. Assume that \( S \) consists of \( n_s \) solvers, \( P \) consists of \( n_p \) problems. For each problem \( p \in P \) and solver \( s \in S \), denote \( t_{p,s} \) be the computing time (or the number of iterations) required to solve problem \( p \in P \) by solver \( s \in S \), and the comparison between different solvers is based on the performance ratio defined by

\[ r_{p,s} = \frac{t_{p,s}}{\min \{ t_{p,s} : s \in S \}}. \]
| Problems     | n     | hPRPHZ | PRP  | HZ  |
|--------------|-------|--------|------|-----|
|              | time  | iter   | time | iter | time  | iter   | time  | iter   |
| FLETCHCR     | 5000  | 95.6800| 34677| 123.9500| 456454| 84.2000| 40000 |
| CURLY30      | 1000  | 8.8600 | 15122| 8.8700  | 15401 | NaN   | NaN   |
| CURLY20      | 1000  | 10.9100| 15084| 6.9600  | 15797 | NaN   | NaN   |
| DIXMAANI     | 6000  | 9.4300 | 2661 | 9.0600  | 2261  | 13.9800| 4720  |
| EIGENBLS     | 420   | 3.5500 | 4978 | 10.1100 | 5440  | 14.9300| 9714  |
| TRIDIA       | 1000  | 7.3200 | 1116 | 3.1900  | 1116  | 3.8900 | 2231  |
| NONDQUAR     | 5000  | 4.2400 | 5099 | 7.5000  | 5085  | 9.4700 | 10058 |
| CURLY10      | 1000  | 4.2700 | 14406| 4.0600  | 13659 | NaN   | NaN   |
| EIGENCLS     | 462   | 4.2500 | 1802 | 4.1000  | 1883  | 5.9900 | 3312  |
| SPARSINE     | 1000  | 2.5700 | 4516 | 4.3200  | 4483  | 6.5000| 8793  |
| EIGENALS     | 420   | 3.9700 | 1344 | 2.4900  | 1306  | 4.7400| 2998  |
| FLETCHCR     | 1000  | 6.0300 | 7479 | 4.9300  | 9139  | 3.5700| 9798  |
| GENHUMPS     | 1000  | 2.2400 | 3555 | 5.8400  | 3435  | 7.5500| 5807  |
| FMINSURF     | 5625  | 1.0000 | 492  | 3.4700  | 669   | 3.3900| 949   |
| TRIDIA       | 5000  | 1.0900 | 783  | 1.0700  | 783   | 1.3100| 1565  |
| DIXMAANE     | 6000  | 1.2200 | 303  | 1.2600  | 306   | 2.1300| 620   |
| DIXMAANJ     | 6000  | 23.8000| 296  | 1.1800  | 275   | 2.1700| 557   |
| BDQRTIC      | 5000  | 1.3500 | 8726 | 7.6400  | 2428  | NaN   | NaN   |
| DIXMAANK     | 6000  | 1.8100 | 264  | 1.1100  | 248   | 1.8000| 587   |
| NONCVXU2     | 1000  | 1.5600 | 2055 | 1.9200  | 2015  | 3.6400| 3919  |
| DIXMAANL     | 6000  | 0.9700 | 245  | 1.3200  | 215   | 3.0100| 702   |
| SENSORS      | 100   | 1.0700 | 44   | 0.9700  | 45    | 1.3600| 66    |
| DIXMAANF     | 6000  | 1.0400 | 230  | 1.1200  | 230   | 1.6200| 437   |
| DIXMAANG     | 6000  | 1.3400 | 227  | 1.0800  | 227   | 1.4500| 420   |
| DIXMAANH     | 6000  | 0.9900 | 224  | 1.1600  | 224   | 2.6400| 825   |
| FLETCHV2     | 1000  | 1.4000 | 1055 | 1.0000  | 1044  | 1.2900| 1886  |
| SCHMVETT     | 10000 | 2.3800 | 60   | 1.5000  | 64    | 2.5900| 105   |
| GENHUMPS     | 500   | 1.0100 | 2258 | 2.1500  | 2531  | 2.7000| 4147  |
| CRAGGLVY     | 5000  | 0.7400 | 143  | 0.9900  | 138   | NaN   | NaN   |
| MOREBV       | 10000 | 1.1900 | 97   | 0.8900  | 97    | 1.2800| 201   |
| WOODS        | 10000 | 0.8400 | 257  | 1.1700  | 230   | 2.1400| 487   |
| NONDQUAR     | 10000 | 0.3800 | 3147 | 1.4500  | 4900  | 1.6300| 8128  |
| SPARSQUR     | 10000 | 0.3500 | 23   | 0.3800  | 23    | 1.1300| 131   |
| POWER        | 5000  | 0.6500 | 259  | 0.6100  | 408   | 0.4000| 514   |
| MANCINO      | 100   | 0.3500 | 12   | 0.6000  | 11    | 1.1500| 27    |
| CRAGGLVY     | 2000  | 0.3300 | 132  | 0.3700  | 142   | NaN   | NaN   |
| CURLY30      | 200   | 0.4800 | 2819 | 0.3600  | 3066  | NaN   | NaN   |
| LIARWHD      | 10000 | 0.5700 | 41   | 0.4600  | 39    | 0.4800| 46    |
| BDQRTIC      | 10000 | 0.4600 | 1025 | 0.4900  | 798   | NaN   | NaN   |
| GENROSE      | 500   | 0.2900 | 1309 | 0.4900  | 1624  | 0.4600| 2278  |
| VARDIM       | 10000 | 0.2700 | 62   | 0.2900  | 57    | NaN   | NaN   |
| CURLY20      | 200   | 0.7100 | 2951 | 0.3000  | 2835  | NaN   | NaN   |
| FREUROTH     | 5000  | 0.4000 | 96   | 0.5900  | 76    | NaN   | NaN   |
| ENGVAL1      | 10000 | 0.2800 | 35   | 0.4100  | 34    | NaN   | NaN   |
| POWELLSG     | 10000 | 0.2500 | 77   | 0.2300  | 49    | 0.7200| 362   |
| DIXON3DQ     | 10000 | 0.3100 | 1002 | 0.2700  | 1002  | 0.3300| 2005  |
| Problems  | \( n \) | \( \text{hPRPHZ} \) | \( \text{PRP} \) | \( \text{HZ} \) |
|-----------|-------|----------------|----------------|----------------|
|           | time  | iter           | time  | iter           | time  | iter           |
| BRYBND    | 5000  | 0.4500         | 39    | 0.3200         | 40    | 0.3800         | 66    |
| HILBERTA  | 200   | 0.7100         | 50    | 0.3700         | 25    | 0.3800         | 38    |
| TQUARTIC  | 10000 | 0.1900         | 61    | 0.6500         | 52    | 0.5800         | 38    |
| CURLY10   | 200   | 0.2100         | 3100  | 0.2000         | 3182  | NaN            | NaN   |
| FLETCBV2  | 500   | 0.2600         | 480   | 0.2200         | 482   | 0.3600         | 962   |
| FMINSURF  | 1024  | 0.1200         | 238   | 0.2400         | 300   | 0.2800         | 455   |
| VARDIM    | 5000  | 0.2000         | 44    | 0.1300         | 47    | NaN            | NaN   |
| FMINSRF2  | 1024  | 0.1400         | 282   | 0.2600         | 355   | 0.2900         | 517   |
| SPMRTLS   | 1000  | 0.2400         | 151   | 0.1500         | 151   | 0.2000         | 281   |
| LIARWHD   | 5000  | 0.2600         | 32    | 0.3000         | 48    | 0.2500         | 46    |
| NONDIA    | 10000 | 0.2600         | 16    | 0.2300         | 10    | 0.3100         | 26    |
| POWELLSG  | 5000  | 0.5500         | 187   | 0.1100         | 53    | 0.3200         | 346   |
| ARWHEAD   | 10000 | 0.1600         | 15    | 0.5300         | 12    | NaN            | NaN   |
| SROSENBR  | 10000 | 0.1900         | 17    | 0.1700         | 19    | 0.1700         | 26    |
| TQUARTIC  | 5000  | 0.1700         | 38    | 0.2100         | 54    | 0.1700         | 32    |
| PENALTY1  | 5000  | 0.2500         | 62    | 0.2200         | 80    | 0.3400         | 152   |
| DQDRTIC   | 10000 | 0.1300         | 8     | 0.2600         | 8     | 0.2700         | 15    |
| NONDIA    | 5000  | 0.2200         | 22    | 0.1400         | 26    | 0.1300         | 26    |
| ARGLINB   | 300   | 0.1300         | 23    | 0.2000         | 17    | NaN            | NaN   |
| DIXMAAND  | 6000  | 0.2500         | 13    | 0.1300         | 12    | 0.1600         | 25    |
| ARGLINC   | 300   | 0.0800         | 19    | 0.2700         | 25    | NaN            | NaN   |
| DQRTIC    | 5000  | 0.0900         | 34    | 0.1000         | 34    | 0.1000         | 66    |
| QUARTC    | 5000  | 0.0900         | 34    | 0.0900         | 34    | 0.1000         | 66    |
| EIGENALS  | 110   | 0.0400         | 389   | 0.0800         | 359   | 0.1600         | 806   |
| SINQUAD   | 500   | 0.0800         | 111   | 0.0400         | 93    | NaN            | NaN   |
| SPARSINE  | 200   | 0.0600         | 445   | 0.0800         | 445   | 0.1300         | 917   |
| DIXON3DQ  | 500   | 0.2400         | 500   | 0.0600         | 500   | 0.0800         | 1003  |
| DIXMAANC  | 6000  | 0.2200         | 11    | 0.2400         | 11    | 0.2600         | 23    |
| HILBERTB  | 200   | 0.2100         | 6     | 0.2200         | 6     | 0.2500         | 13    |
| BROWNAL   | 400   | 0.0700         | 13    | 0.2000         | 7     | 0.2700         | 37    |
| EIGENCLS  | 90    | 0.2500         | 360   | 0.0700         | 350   | 0.1100         | 743   |
| ARGLINA   | 300   | 0.2300         | 2     | 0.2500         | 2     | 0.2600         | 5     |
| EXTRONSNB | 50    | 0.1300         | 5819  | 0.1900         | 5294  | 0.2400         | 7808  |
| PENALTY2  | 200   | 0.1800         | 365   | 0.1400         | 417   | NaN            | NaN   |
| FREUROTH  | 1000  | 0.0700         | 187   | 0.1600         | 137   | NaN            | NaN   |
| BRYBND    | 1000  | 0.0600         | 52    | 0.0600         | 35    | 0.0800         | 73    |
| DIXMAANB  | 3000  | 0.0400         | 10    | 0.0600         | 10    | 0.0700         | 23    |
| NONCVUX2  | 100   | 0.0600         | 396   | 0.0300         | 414   | 0.0500         | 801   |
| DIXMAANA  | 3000  | 0.2100         | 10    | 0.0500         | 9     | 0.0700         | 20    |
| TOINTGSS  | 10000 | 0.0300         | 5     | 0.2100         | 5     | 0.3800         | 20    |
| POWER     | 1000  | 0.0600         | 117   | 0.0600         | 222   | 0.0400         | 236   |
| DECONVU   | 61    | 0.0200         | 462   | 0.0600         | 460   | 0.0700         | 581   |
| GENROSE   | 100   | 0.0200         | 347   | 0.0200         | 392   | 0.0300         | 626   |
| COSINE    | 1000  | 0.0300         | 24    | 0.0200         | 24    | 0.0300         | 29    |
| DIXMAANB  | 1500  | 0.0100         | 10    | 0.0300         | 10    | 0.0400         | 24    |
| CHNROSNB  | 50    | 0.0300         | 273   | 0.0200         | 285   | 0.0100         | 500   |
| Problems     | n    | hPRPHZ time | hPRPHZ iter | PRP time | PRP iter | HZ time | HZ iter |
|--------------|------|-------------|-------------|----------|----------|---------|---------|
| DIXMAANA     | 1500 | 0.0100      | 10          | 0.0300   | 9        | 0.0300  | 22      |
| FMINSRF2     | 121  | 0.0300      | 115         | 0.0100   | 124      | 0.0100  | 250     |
| ARWHEAD      | 1000 | 0.0100      | 16          | 0.0300   | 19       | NaN     | NaN     |
| COSINE       | 50   | 0.0200      | 23          | 0        | 22       | 0.0100  | 26      |
| DQDRTIC      | 1000 | 0.0600      | 8           | 0.0200   | 8        | 0.0300  | 15      |
| ERRINROS     | 50   | 0.0200      | 1444        | 0.0900   | 2416     | NaN     | NaN     |
| EG2          | 1000 | 0.0100      | 6           | 0.0100   | 6        | NaN     | NaN     |
| TESTQUAD     | 100  | 0.0100      | 321         | 0.0100   | 303      | 0.0100  | 925     |
| TOINTGOR     | 50   | 0.8800      | 151         | 0.0100   | 155      | 0.0100  | 250     |
| SPARSINE     | 5000 | 0.1300      | 370         | 1.5700   | 544      | 1.1200  | 719     |
| FMINSRF2     | 10000| 0.2800      | 26          | 0.1200   | 23       | 0.1300  | 27      |
| FMINSRF2     | 15625| 1.1300      | 28          | 0.2600   | 23       | 0.2800  | 28      |
| FMINSRF2     | 5625 | 3.0500      | 227         | 1.3100   | 214      | 1.8900  | 430     |
| NONDQUAR     | 10000| 1.3200      | 234         | 2.4200   | 225      | 3.5200  | 440     |
| POWER        | 10000| 43.7500     | 142         | 0.7100   | 62       | NaN     | NaN     |
| ARWHEAD      | 5000 | 0.2100      | 7298        | 36.8400  | 6398     | NaN     | NaN     |
| COSINE       | 5000 | 59.1900     | 37          | 0.2000   | 35       | NaN     | NaN     |
| COSINE       | 10000| 3.6600      | 8476        | 31.5200  | 4721     | 53.2500 | 8965    |
| FMINSURF     | 10000| 0.6400      | 8771        | 2.1400   | 5022     | 2.4100  | 6779    |
| FMINSURF     | 15625| 0.3600      | 108         | 0.4700   | 62       | NaN     | NaN     |
| BROYDN7D     | 10000| 5.3900      | 498         | 0.2700   | 371      | NaN     | NaN     |
| SPMSRRTLS    | 4999 | 0.0010      | 2232        | 5.4700   | 2183     | 6.4500  | 4093    |
| SPMSRRTLS    | 10000| 0.0010      | NaN         | NaN      | NaN      | 0.2800  | NaN     |
| FREUROTH     | 10000| 0.0010      | NaN         | NaN      | NaN      | 1.8900  | NaN     |
| FLETCBV2     | 500  | 0.0010      | NaN         | NaN      | NaN      | 3.5200  | NaN     |
| BDQRTIC      | 10000| 0.0010      | 1           | NaN      | NaN      | 0.2800  | NaN     |
| VAREIGVL     | 10000| 0.0010      | 1           | NaN      | NaN      | 1.8900  | NaN     |
| ENGVAL1      | 5000 | NaN         | 1           | NaN      | NaN      | 3.5200  | NaN     |
| BRYBND       | 10000| 0.1000      | 34677       | 0.5000   | 456454   | 0.9000  | 40000   |
| EIGENBLS     | 930  | 0.1000      | 15122       | 0.0500   | 15401    | 0.9000  | NaN     |
| NONCVXUN     | 500  | 0.1000      | 15084       | 0.0500   | 15797    | 0.9000  | NaN     |
| GENROSE      | 10000| 0.1000      | 2661        | 0.5000   | 2261     | 0.9000  | 4720    |
| GENROSE      | 5000 | 0.1000      | 4978        | 0.0500   | 5440     | 0.9000  | 9714    |
| EIGENALS     | 930  | 0.1000      | 1116        | 0.0500   | 1116     | 0.9000  | 2231    |
| SINQUAD      | 5000 | 0.1000      | 5099        | 0.5000   | 5058     | 0.9000  | 10058   |
| SINQUAD      | 10000| 0.1000      | 14406       | 0.0500   | 13659    | 0.9000  | NaN     |
| GENHUMPS     | 5000 | 0.1000      | 1802        | 0.0500   | 1883     | 0.9000  | 3312    |
| CHAINWOO     | 10000| 0.1000      | 4516        | 0.5000   | 4483     | 0.9000  | 8793    |
| TESTQUAD     | 10000| 0.1000      | 1344        | 0.0500   | 1306     | 0.9000  | 2998    |
| TESTQUAD     | 10000| 0.1000      | 7479        | 0.0500   | 9139     | 0.9000  | 8986    |
| TESTQUAD     | 5000 | 0.1000      | 3555        | 0.5000   | 3435     | 0.9000  | 5807    |
| FLETCHCR     | 5000 | 0.1000      | 492         | 0.0500   | 669      | 0.9000  | 949     |
| CURLY30      | 10000| 0.1000      | 783         | 0.0500   | 783      | 0.9000  | 1565    |
| CURLY20      | 10000| 0.1000      | 303         | NaN      | 306      | 0.9000  | 620     |
| DIXMAANI      | 6000 | 0.1000      | 296         | NaN      | 275      | 0.9000  | 557     |
| EIGENBLS     | 420  | 0.1000      | 8726        | NaN      | 2428     | 0.9000  | NaN     |
Assume that a large enough parameter \( r_M \geq r_{p,s} \) for all \( p, s \) is chosen, and \( r_{p,s} = r_M \) if and only if solvers \( s \) does not solver problem \( p \). Define

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

where size \( A \) means the number of elements in set \( A \), then \( \rho_s(\tau) \) is the probability for solver \( s \in S \) that a performance ratio \( r_{p,s} \) is within a factor \( \tau \in \mathbb{R}^n \). The \( \rho_s \) is the (cumulative) distribution function for the performance ratio. The value of \( \rho_s(1) \) is the probability that the solver will win over the rest of the solvers.

That is, for each method, we plot the fraction \( P \) of problems for which the method is within a factor of the best time. The left side of the figure gives the percentage of the test problems for which a method is the fastest; the right side gives the percentage of the test problems that are successfully solved by each of the methods. The top curve is the method that solved the most problems in a time that was within a factor of the best time.

Based on the theory of the performance profile above, four performance figures, i.e., Figs. 1–2 can be generated according to Table 1.

From the four figures, we can see that the hPRPHZ is superior to the other conjugate gradient methods on the testing problems.

![Performance Profile based on the iteration number.](image)

**Figure 1.**

5. **Conclusion.** In this work, we proposed a new conjugate gradient method for unconstrained optimization, where the parameter \( \beta_k \) computed as a convex combination of HZ and PRP. The sufficient descent and global convergence was proved and numerical performance support the effectiveness and robustness of our procedure.
Figure 2.

REFERENCES

[1] M. Al-Baali, Descent property and global convergence of the Fletcher–Reeves method with inexact line search, IMA Journal of Numerical Analysis, 5 (1985), 121–124.
[2] N. Andrei, Another hybrid conjugate gradient algorithm for unconstrained optimization, Numerical Algorithms, 47 (2008), 143–156.
[3] N. Andrei, New hybrid conjugate gradient algorithms for unconstrained optimization, in Encyclopedia of Optimization (eds. A. F. Christodoulou and M. P. Panos), 2009.
[4] B. Balaram, M. Narayanan and P. Rajendrakumar, Optimal design of multi-parametric nonlinear systems using a parametric continuation based genetic algorithm approach, Nonlinear Dynamics, 67 (2012), 2759–2777.
[5] I. Bongartz, A. R. Conn, N. Gould and P. L. Toint, CUTE: constrained and unconstrained testing environments, ACM Transactions on Mathematical Software (TOMS), 21 (1995), 123–160.
[6] N. Chenna, Comments on “New Hybrid Conjugate Gradient Method as a Convex Combination of FR and PRP Methods”, FILOMAT, 33 (2019), 3083–3100.
[7] Y.-H. Dai and Y. Yuan, A nonlinear conjugate gradient method with a strong global convergence property, SIAM Journal on Optimization, 10 (1999), 177–182.
[8] S. S. Djordjević, New hybrid conjugate gradient method as a convex combination of FR and PRP methods, Filomat, 30 (2016), 3083–3100.
[9] E. D. Dolan and J. J. Moré, Benchmarking optimization software with performance profiles, Mathematical Programming, 91 (2002), 201–213.
[10] 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.
[11] X. Z. Jiang, G.-D. Ma and J.-B. Jian, A new global convergent conjugate gradient method with Wolfe line search, Chinese Journal of Engineering Mathematics, 28 (2011), 779–786.
[12] J. Liu and S. Li, New hybrid conjugate gradient method for unconstrained optimization, Applied Mathematics and Computation, 245 (2014), 36–43.
[13] E. Polak and G. Ribière, Note sur la convergence de méthodes de directions conjuguées, Revue Française D’informatique et De Recherche Opérationnelle, Série Rouge, 3 (1969), 35–43.
[14] B. T. Polyak, The conjugate gradient method in extremal problems, *USSR Computational Mathematics and Mathematical Physics*, 9 (1969), 94–112.

[15] M. Powell, Nonconvex minimization calculations and the conjugate gradient method, *Numerical Analysis* (ed. D. F. Griffiths), Volume 1066 of Lecture Notes in Math., Dundee, (1984), 122–141.

[16] Z. Wei, G. Li and L. Qi, New nonlinear conjugate gradient formulas for large-scale unconstrained optimization problems, *Applied Mathematics and Computation*, 179 (2006), 407–430.

[17] G. Yuan, Z. Wei and Q. Zhao, A modified Polak–Ribiè re–Polyak conjugate gradient algorithm for large-scale optimization problems, *IIE Transactions*, 46 (2014), 397–413.

[18] G. Zoutendijk, Nonlinear programming, computational methods, *Integer and Nonlinear Programming*, (1970), 37–86.

Received January 2020; 1st revision February 2020; Final revision April 2020.

E-mail address: dell.publication@gmail.com
E-mail address: m.belloufi@univ-Soukahras.dz
E-mail address: bsellami@univ-soukahras.dz