Journal of Mathematical Modeling
Vol. 12, No. 2, 2024, pp. 301–317. Research Article

JMM

A hybrid CG algorithm for nonlinear unconstrained optimization with application in image restoration

Choubeila Souli[1], Raouf Ziadi[1]*, Abdelatif Bencherif-Madani[1], Hisham Mohammed Khudhur[2]

[1]*Laboratory of Fundamental and Numerical Mathematics (LMFN), University Ferhat Abbas Setif 1, Algeria*
[2]*Department of Mathematics, College of Computer Science and Mathematics, University of Mosul, Mosul, Iraq*
Email(s): souli.choubeila25@gmail.com, ziadi.raouf@gmail.com, lotfi_madani@yahoo.fr, hisham892020@uomosul.edu.iq

Abstract. This paper presents a new hybrid conjugate gradient method for solving nonlinear unconstrained optimization problems; it is based on a combination of *RMIL* (Rivaie-Mustafa-Ismail-Leong) and *hSM* (hybrid Sulaiman-Mohammed) methods. The proposed algorithm enjoys the sufficient descent condition without depending on any line search; moreover, it is globally convergent under the usual and strong Wolfe line search assumptions. The performance of the algorithm is demonstrated through numerical experiments on a set of 100 test functions from [1] and four image restoration problems with two noise levels. The numerical comparisons with four existing methods show that the proposed method is promising and effective.

Keywords: Unconstrained optimization, Hybrid conjugate gradient, Global convergence, Image restoration.
AMS Subject Classification 2010: 90C26, 90C30

1 Introduction

In this work, we consider the following unconstrained problem

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

where $f:\mathbb{R}^n \longrightarrow \mathbb{R}$ is a continuously differentiable function with available gradient $\nabla f(x)$. The problem (P) is of interest in many real-world applications involving objective functions which are continuously differentiable [22]. To mention just of few, these are applied to molecular physics [21, 23], statistical modelling [13, 18] and image processing [8]. The conjugate gradient methods (CG) are among the most effective methods for solving the problems of type (P) due to their simplicity and low storage. They take up several forms; their principle is to generate a sequence of points $\{x_k\}_{k\geq 0} \subset \mathbb{R}^n$ starting from an initial point $x_0 \in \mathbb{R}^n$ following the procedure

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

where d_k is a descent direction for f at x_k and $\alpha_k \in \mathbb{R}^+$ is a step-length which ensures that x_{k+1} is a feasible point with $f(x_{k+1}) \leq f(x_k)$. The step-length α_k in (1) is determined by using a line search procedure which ensures that

*Corresponding author

Received: 28 November 2023 / Revised: 5 February 2024 / Accepted: 13 February 2024
DOI: 10.22124/JMM.2024.26151.2317

the sufficient decrease conditions are satisfied at the new point x_{k+1}; typically, it is chosen in such a way that it satisfies the weak Wolfe conditions

$$\begin{aligned} f(x_k+\alpha_k d_k)-f(x_k) &\leq \delta \alpha_k g_k^T d_k, \\ \nabla f(x_k+\alpha_k d_k)^T d_k &\geq \sigma g_k^T d_k, \end{aligned} \tag{2}$$

or the strong Wolfe conditions

$$\begin{aligned} f(x_k+\alpha_k d_k)-f(x_k) &\leq \delta \alpha_k g_k^T d_k, \\ |\nabla f(x_k+\alpha_k d_k)^T d_k| &\leq -\sigma g_k^T d_k, \end{aligned} \tag{3}$$

where $\delta \in (0,1/2)$, $\sigma \in (\delta,1)$ and $g_k = \nabla f(x_k)$. The search direction d_k is usually defined as

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

where the parameter β_k is a scalar which determines the different conjugate gradient methods. There are several well-known β_k formulas including Hestenes-Stiefel parameter [7] with $\beta_k^{HS} = \frac{g_k^T y_{k-1}}{d_{k-1}^T y_{k-1}}$, Fletcher-Reeves [5] with $\beta_k^{FR} = \frac{\|g_k\|^2}{\|g_{k-1}\|^2}$, Polyak-Polak-Ribière [15] with $\beta_k^{PRP} = \frac{g_k^T y_{k-1}}{\|g_{k-1}\|^2}$, Conjugate-Descent [4] with $\beta_k^{CD} = -\frac{\|g_k\|^2}{g_{k-1}^T d_{k-1}}$, Liu-Storey [10] with $\beta_k^{LS} = -\frac{g_k^T y_{k-1}}{g_{k-1}^T d_{k-1}}$ and Dai-Yaun parameter [2] with $\beta_k^{DY} = \frac{\|g_k\|^2}{d_{k-1}^T y_{k-1}}$, where $\|.\|$ is the Euclidean norm in $\mathbb{R}^n$ and $y_{k-1} = g_k - g_{k-1}$. In the case of a strictly convex function with an exact line search all the variants mentioned above are equivalent, but they behave differently for non-linear objective functions using inexact line searches.

The most important characteristics of CG methods are their global convergence and numerical performances. According to [2, 4, 5, 7, 10, 15, 16] the methods mentioned above form two classes. The FR, CD and DY methods have excellent global convergence and not so good practical behaviour. On the contrary, the HS, PRP and LS have superior numerical performances, but they may not always converge. New hybrid conjugate gradient methods that combine a good practical performance and powerful global convergence properties have been suggested in the literature. The first one was proposed by Touati-Ahmed and Storey [19] where the parameter β_k is chosen as follows

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

Using the strong Wolfe line search, this TS method enjoys both the good convergence results of the FR algorithm and the satisfactory numerical performance of the PRP method. Koontse and Kaelo [9] have also proposed an interesting hybrid conjugate gradient where the parameter β_k is given by

$$\beta_k^* = \max\{\min\{-c\beta_k^{PRP}, \beta_k^{FR}\}, \min\{\beta_k^{FR}, \beta_k^{PRP}\}\}, \quad c = \frac{1-\gamma}{1+\gamma}, \quad \gamma \in [1/2, 1],$$

and the search direction is

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

where $\xi_k = 1 + \beta_k \frac{d_{k-1}^T g_k}{\|g_k\|^2}$. A modification of the PRP method, called RMIL, was proposed by Rivaie et al. [17]. This hybrid CG method uses the PRP numerator and β_k is defined as follows

$$\beta_k^{RMIL} = \frac{g_k^T (g_k - g_{k-1})}{\|d_{k-1}\|^2}; \tag{4}$$

it has a restart condition with a good convergence and an effective numerical performance. Sulaiman et al. [18] proposed a hybrid CG method where β_k is defined as

$$\beta_k^{hSM^*} = \begin{cases} \beta_k^{RMIL}, & \text{if } 0 \le \beta_k^{RMIL} \le \beta_k^{hSM}, \\ \beta_k^{hSM}, & \text{otherwise }, \end{cases}$$

with

$$\beta_k^{hSM} = \frac{g_k^T(g_k + g_{k-1})}{\|d_{k-1}\|^2}. \tag{5}$$

The authors proved the global convergence under strong Wolfe line search conditions and their numerical experiments show that hSM* method is competitive. To achieve both effectiveness and robust convergence characteristics, several hybrid conjugate gradient methods based on the concept of convex combinations have been proposed in the literature. Livieris et al. [11] suggested two hybrid conjugate gradient methods named ADHCG1 and ADHCG2, where the scalars β_k^{ADHCG1} and β_k^{ADHCG2} are defined as follow

$$\beta_k^{ADHCG(i)} = \theta_k^{(i)}\beta_k^{DY} + (1-\theta_k^{(i)})\beta_k^{HS^+}, \quad i = 1,2$$

where $\theta_k^{(1)}, \theta_k^{(2)} \in [0,1]$ and $\beta_k^{HS^+} = \max\{0, \beta_k^{HS}\}$. The authors compute the hybridization parameters $\theta_k^{(1)}$ and $\theta_k^{(2)}$ by minimizing the distance between the hybrid CG direction and the self-scaling memoryless BFGS direction where the search directions $d_k^{(i)}, i = 1,2$ are as follows

$$d_k^{(i)} = -(1 + \beta_k^{ADHCG(i)}\frac{g_k^T d_{k-1}}{\|g_k\|})g_k + \beta_k^{ADHCG(i)} d_{k-1}, \quad i = 1,2.$$

Mtagulwa and Kaelo [14] have also introduced a hybrid CG method named EPF based on a convex combination of NPRP [20] and FR methods in which the parameter β_k is defined as

$$\beta_k^{EPF} = \begin{cases} \beta_k^{PRP}, & \text{if} \quad \|g_k\|^2 > |g_k^T g_{k-1}|, \\ \theta_k\beta_k^{FR} + (1-\theta_k)\beta_k^{NPRP}, & \text{otherwise}, \end{cases}$$

where $\theta_k \in [0,1]$,

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

and the search direction is computed as

$$d_k = -g_k + \beta_k^{EPF} d_{k-1} - \beta_k^{EPF}\frac{d_{k-1}^T g_k}{\|g_k\|^2} g_k.$$

Recently, Lotfi and Hosseini [12] have introduced a hybrid CG method named THCG+, based on a convex combination of PRP$^+$ [6] and FR methods in which the parameter β_k is defined as

$$\beta_k^{THCG+} = \theta_k\beta_k^{FR} + (1-\theta_k)\beta_k^{PRP^+},$$

where $\theta_k \in [0,1]$, $\beta_k^{PRP^+} = \max\{0, \beta_k^{PRP}\}$ and the search direction is taken as

$$d_k^{THCG+} = -g_k + \beta_k^{THCG+} d_{k-1} - \beta_k^{THCG+}\frac{g_k^T d_{k-1}}{\|g_k\|^2} g_k.$$

Inspired by the methods mentioned above, we propose a new hybrid conjugate gradient method, named CR (Combined method for Restoring images) method, to simultaneously solve unconstrained optimization and also deal with image restoration problems. The suggested method possesses the following properties:

- It integrates the features of the RMIL and hSM methods.
- The generated search direction satisfies the sufficient descent condition independently of line searches, the global convergence is proved under the usual mild conditions and the strong Wolfe line search.
- The numerical performance is efficient and image restoration is successful.

The rest of this paper is organized as follows. We describe the algorithm in detail in Section 2. Next, in Section 3, we study the sufficient descent condition and the global convergence. In the last Section, numerical results are reported and some conclusions are drawn.

2 The proposed algorithm

The parameter β_k is regarded as a convex combination as follows

$$\beta_k^{CR} = (1-\theta_k)\beta_k^{RMIL} + \theta_k \beta_k^{hSM}, \tag{6}$$

where $\theta_k \in [0,1]$, β_k^{RMIL} and β_k^{hSM} are defined in equations (4) and (5) respectively. To ensure our method generates descent directions which enhance computational efficiency and robustness, we compute the search direction d_k as

$$d_k = \begin{cases} -g_k, & \text{if} \quad k=0, \\ -g_k + \beta_k^{CR}(d_{k-1} - \rho_k g_k), & \text{if} \quad k \geq 1, \end{cases} \tag{7}$$

where $\rho_k = \frac{d_{k-1}^T g_k}{\|g_k\|^2}$. Therefore, for $k \geq 1$

$$d_k = -g_k + \beta_k^{CR}(d_{k-1} - \rho_k g_k) = -g_k + (\beta_k^{RMIL} + \theta_k(\beta_k^{hSM} - \beta_k^{RMIL}))(d_{k-1} - \rho_k g_k),$$

that is,

$$d_k = -g_k + (\beta_k^{RMIL} + \theta_k \frac{2 g_k^T g_{k-1}}{\|d_{k-1}\|^2})(d_{k-1} - \rho_k g_k). \tag{8}$$

Now, we need to consider the parameter θ_k. It is selected such that the search direction d_k satisfies also the following conjugacy condition

$$y_{k-1}^T d_k = 0. \tag{9}$$

From (8) and (9), we get

$$0 = -y_{k-1}^T g_k + (\beta_k^{RMIL} + \theta_k \frac{2 g_k^T g_{k-1}}{\|d_{k-1}\|^2})(y_{k-1}^T d_{k-1} - \rho_k y_{k-1}^T g_k),$$

then

$$\theta_k = \frac{y_{k-1}^T g_k - \beta_k^{RMIL}(y_{k-1}^T d_{k-1} - \rho_k y_{k-1}^T g_k)}{2\frac{g_k^T g_{k-1}}{\|d_{k-1}\|^2}(y_{k-1}^T d_{k-1} - \rho_k y_{k-1}^T g_k)} = \frac{\zeta_k - \beta_k^{RMIL}\lambda_k}{\eta_k \lambda_k}, \tag{10}$$

where $\zeta_k = y_{k-1}^T g_k$, $\lambda_k = y_{k-1}^T d_{k-1} - \rho_k y_{k-1}^T g_k$ and $\eta_k = 2\frac{g_k^T g_{k-1}}{\|d_{k-1}\|^2}$.

During the search process, if for such an iteration we have $\eta_k \lambda_k = 0$ or $\theta_k < 0$ we set $\theta_k = 0$; and in the case where $\theta_k > 1$ we set $\theta_k = 1$. Algorithm 1 below summarizes the main steps of the proposed method.

Algorithm 1 The CR algorithm.

0. (Initialization) Select $x_0 \in \mathbb{R}^n$ and the parameters $0 < \delta < \sigma < 1$, $\varepsilon > 0$. Compute $f(x_0)$, $g_0 = \nabla f(x_0)$ and $d_0 = -g_0$. Set $k = 0$.

1. If $\|g_k\| \leq \varepsilon$, then stop; otherwise:
- Compute the step-length $\alpha_k > 0$ along the direction d_k using the strong Wolfe line search technique (3).
- Put $x_{k+1} = x_k + \alpha_k d_k$ and set $k = k+1$.

2. Compute the parameter θ_k: if $\eta_k \lambda_k = 0$ put $\theta_k = 0$, otherwise compute θ_k following the equation (10).

3. β_k **computation:** β_k is computed following the equation (6) for $\theta_k \in (0,1)$; otherwise set $\beta_k = \beta_k^{RMIL}$ if $\theta_k \leq 0$ and, if $\theta_k \geq 1$, set $\beta_k = \beta_k^{hSM}$.

4. **Search direction computation:** if the restart criterion of Powell $|g_k^T g_{k-1}| \geq 0.2\,\|g_k\|^2$ holds, set $d_k = -g_k$; otherwise d_k is computed as in (7) and repeat Step 1.

3 The sufficient descent condition and the global convergence of CR method

3.1 The sufficient descent condition

It is well known that the sufficient descent property is crucial for the global convergence to hold. The next lemma deals with this issue, moreover it shows it is independent of any line search.

Lemma 1. *Let the sequences* $\{g_k\}_{k\in\mathbb{N}}$ *and* $\{d_k\}_{k\in\mathbb{N}}$ *be given by CR algorithm, then*

$$g_k^T d_k = -\|g_k\|^2, \tag{11}$$

i.e., the direction d_k *satisfies the sufficient descent condition.*

Proof. It is clear that for $k = 0$, the equation (11) is satisfied, that is $g_0^T d_0 = -\|g_0\|^2$. Now for $k \geq 1$, we have

$$d_k = -g_k + \beta_k^{CR}(d_{k-1} - \rho_k g_k),$$

taking an inner product with g_k^T we get

$$g_k^T d_k = -\|g_k\|^2 - \beta_k^{CR} g_k^T d_{k-1} + \beta_k^{CR} g_k^T d_{k-1} = -\|g_k\|^2,$$

which finishes the proof. □

3.2 The global convergence

We require the following mild assumptions.

Assumption 1. *The level set* $\Omega = \{x \in \mathbb{R}^n : f(x) \leq f(x_0)\}$ *is a bounded, i.e. there exists a constant* $K > 0$ *such that* $\|x\| \leq K, \quad \forall x \in \Omega$.

Assumption 2. *In a close neighbourhood* $\mathcal{N}$ *of* Ω*, the objective function* f *is continuously differentiable and its gradient* g *is Lipschitz continuous, i.e., there exists a positive constant* L *such that*

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

Note that Assumptions 1 and 2 imply that there exists a positive constant $\bar{\delta}$ such that

$$\|g(x)\| \leq \bar{\delta}, \quad \forall x \in \mathcal{N}.$$

Lemma 2. *Assume that Assumptions 1 and 2 hold. If the step-length α_k satisfies the strong Wolfe conditions (3) and d_k is a descent direction, then:*

$$\alpha_k \geq \frac{(\sigma-1)}{L}\frac{d_k^T g_k}{\|d_k\|^2}. \tag{12}$$

Proof. From the computation

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

it follows that,

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

□

Note that from (3), (11) and (12), it is clear that $\alpha_k \neq 0$. Hence, a constant $\bar{\gamma} > 0$ must exist such that $\alpha \geq \bar{\gamma} > 0$, for all $k \geq 0$.

In order to prove the global convergence of the CR method, we need the following Lemma, which is due to Zoutendijk [24].

Lemma 3. *Assume that Assumptions 1 and 2 hold and consider any CG method that follows the form (1) where d_k is a descent direction and α_k satisfies the strong Wolfe line search (3). Then the Zoutendijk condition*

$$\sum_{k\geq 0}\frac{(g_k^T d_k)^2}{\|d_k\|^2} < +\infty, \tag{13}$$

holds.

Now, we are in the position to deal with the global convergence result

Theorem 1. *Suppose that Assumptions 1 and 2 hold. Let $\{g_k\}_{k\in\mathbb{N}}$ and $\{d_k\}_{k\in\mathbb{N}}$ be the sequences generated by CR algorithm; then*

$$\liminf_{k\to\infty}\|g_k\| = 0. \tag{14}$$

Proof. Assume that (14) is false, then a constant $C > 0$ exists such that

$$\|g_k\| \geq C, \quad k \in \mathbb{N}. \tag{15}$$

Let $D = \max\{\|x-y\| : x,y \in \mathcal{N}\}$ be the diameter of the level set's neighbourhood $\mathcal{N}$. By the Lipschitz continuity of g we have

$$\|g_k - g_{k-1}\| \leq L\|x_k - x_{k-1}\| \leq LD.$$

From (6), we have

$$|\beta_k^{CR}| = |(1-\theta_k)\beta_k^{RMIL} + \theta_k\beta_k^{hSM}| \leq |\beta_k^{RMIL}| + |\beta_k^{hSM}|.$$

On the other hand, we have also

$$|\beta_k^{RMIL}| \leq \frac{\|g_k\|\|y_{k-1}\|}{\|d_{k-1}\|^2} \leq \frac{\bar{\delta}LD}{B^2} = G1,$$

$$|\beta_k^{hSM}| \leq \frac{\|g_k\|\|g_k + g_{k-1}\|}{\|d_{k-1}\|^2} \leq \frac{\bar{\delta}A}{B^2} = G2,$$

so that

$$|\beta_k^{CR}| \leq G1 + G2 = G. \tag{16}$$

Thus, since $\forall k \in \mathbb{N}, \alpha \geq \bar{\gamma} > 0$, then from (7) and (16) it follows that,

$$\begin{aligned}
\|d_k\| &\leq \|g_k\| + |\beta_k^{CR}|(\|d_{k-1}\| + \|\rho_k\|\|g_k\|) \\
&= \|g_k\| + 2G\|d_{k-1}\| \\
&\leq \bar{\delta} + 2G\frac{\|x_k - x_{k-1}\|}{|\alpha_{k-1}|} \\
&\leq \bar{\delta} + \frac{2GD}{\bar{\gamma}} = E.
\end{aligned}$$

Therefore,

$$\sum_{k\geq 0} \frac{1}{\|d_k\|^2} \geq \frac{1}{E^2} \sum_{k\geq 0} 1 = +\infty. \tag{17}$$

From (11), (13), and (15) we have

$$C^4 \sum_{k\geq 0} \frac{1}{\|d_k\|^2} \leq \sum_{k\geq 0} \frac{\|g_k\|^4}{\|d_k\|^2} = \sum_{k\geq 0} \frac{(g_k^T d_k)^2}{\|d_k\|^2} < +\infty,$$

which is a contradiction, so the assertion (14) is true. □

4 Numerical experiments

We present here a series of numerical results concerning the *CR* method applied on a collection of 34 functions with 100 test problems chosen from [1], as specified in Table 1, using dimensions ranging from 2 to 60000, Image Restoration problems are also considered. All numerical experiments are implemented in the scientific software MATLAB version R2015a, and run on PC with Intel(R) Core i3-4005U CPU 1.70 GHz and 4.00 RAM.

In the first part of this section we compare the performance of the proposed method with seven conjugate gradient methods that are: PRP [15], EPF [14], RMIL [17], hSM* [18], THCG+ [12], ADHCG1 and ADHCG2 [11] (four of them are recent hybrid CG methods: EPF, THCG+, ADHCG1 and ADHCG2). The methods PRP, EPF, RMIL, hSM* and THCG+ are implemented using the strong Wolfe conditions, whereas ADHCG1 and ADHCG2 are implemented using the weak Wolfe conditions, by setting $\delta = 10^{-4}$ and $\sigma = 10^{-3}$. In this comparison, for each test function, the same initial point is chosen for these methods and every computation is terminated when a point x_k satisfying $\|g_k\|_\infty \leq 10^{-6}$ is found within 2000 iterations and whose calculation time does not exceed 500 seconds; otherwise, the computation is considered as a failure.

Throughout the numerical results, in Figures 1-4 we compare the performance of CR method with PRP, EPF, RMIL, hSM*, THCG+, ADHCG1 and ADHCG2 methods using the logarithmic performance profile of Dolan and Moré [3], relative to the number of iterations, function evaluations, gradient evaluations and CPU-time. For a solver s we define the ratio

$$r_{\mathrm{P},s} = \frac{N_{\mathrm{P},s}}{\min\{N_{\mathrm{P},s} : s \in S\}},$$

where $N_{\mathrm{P},s}$ denotes either the number of iterations, number of function (gradient) evaluations, or CPU-time required by the solver s to solve a problem P. If a solver s does not solve the problem P, the ratio $r_{\mathrm{P},s}$ is assigned a large number. The logarithmic performance profile for each solver s is defined as follows

$$\rho_s(\tau) = \frac{\text{number of problems where } \log_2(r_{\mathrm{P},s}) \leq \tau}{\text{total number of problems}},$$

For each method, we plot the fraction $\rho_s(\tau)$ of problems for which the method has a number of iterations (resp. number of function (gradient) evaluations and CPU-time) that is within a factor τ and the top curve in the plot corresponds to the method that solves most problems within a factor τ, for more details see [3].

Table 1: List of test problems.

Function	Initial points	Dimension n	Function	Initial points	Dimension n
Extended White and Holst	(1.1,...,1.1)	4000	Extended Maratos	(2,...,2)	500
Extended White and Holst	(1.1,...,1.1)	5000	Extended Maratos	(2,...,2)	700
Extended White and Holst	(1.1,...,1.1)	6000	Extended Maratos	(2,...,2)	1000
Extended Rosenbrock	(0.5,...,0.5)	1200	Extended Maratos	(2,...,2)	1500
Extended Rosenbrock	(0.5,...,0.5)	3000	POWER	(1,...,1)	2
Extended Rosenbrock	(0.5,...,0.5)	4000	Extended Quadratic Penalty QP1	(1,...,1)	50
Extended Rosenbrock	(0.5,...,0.5)	5000	Extended Quadratic Penalty QP1	(1,...,1)	100
Extended Freudenstein and Roth	(200,...,200)	9000	Extended Quadratic Penalty QP1	(1,...,1)	700
Extended Freudenstein and Roth	(200,...,200)	10000	Extended Quadratic Penalty QP1	(1,...,1)	1000
Extended Freudenstein and Roth	(200,...,200)	20000	Extended Quadratic Penalty QP1	(1,...,1)	1500
Extended Freudenstein and Roth	(200,...,200)	50000	Quadratic QF2	(1,...,1)	5000
Raydan 2	(1,...,1)	1000	Quadratic QF2	(1,...,1)	7000
Raydan 2	(1,...,1)	1500	Quadratic QF2	(1,...,1)	9000
Extended Tridiagonal 1	(2,...,2)	80	Extended Quadratic Penalty QP2	(1,....,1)	40
Extended Tridiagonal 1	(2,...,2)	90	Extended Quadratic Penalty QP2	(1,....,1)	60
Generalized Tridiagonal 1	(2,...,2)	10	Extended Quadratic Penalty QP2	(1,....,1)	70
Generalized Tridiagonal 1	(2,...,2)	20	ENGVAL1	(2,...,2)	1500
Generalized Tridiagonal 1	(2,...,2)	30	ENGVAL1	(2,...,2)	1600
Diagonal 3	(-0.1,...,-0.1)	6	ENGVAL1	(2,...,2)	1800
Diagonal 4	(1,....,1)	30000	Quartic	(0.8,...,0.8)	5000
Diagonal 4	(1,....,1)	40000	Quartic	(0.8,...,0.8)	8000
Diagonal 4	(1,....,1)	50000	Quartic	(0.8,...,0.8)	9000
Diagonal 4	(1,....,1)	60000	Quartic	(0.8,...,0.8)	10000
Diagonal 5	(1.1,...1.1)	1000	HIMMELBH	(0.1,...,0.1)	2000
Diagonal 5	(1.1,...1.1)	1500	HIMMELBH	(0.1,...,0.1)	2500
Diagonal 5	(1.1,...1.1)	2000	HIMMELBH	(0.1,...,0.1)	2700
Diagonal 5	(1.1,...1.1)	2500	HIMMELBH	(0.1,...,0.1)	3000
Diagonal 7	(1,...,1)	700	Extended BD1	(3,....,3)	2000
Diagonal 7	(1,...,1)	1500	Extended BD1	(3,....,3)	3000
Diagonal 7	(1,...,1)	2000	Extended BD1	(3,....,3)	5000
Diagonal 7	(1,...,1)	7000	Extended PSC1	(3,0.1,....,3,0.1)	2
Diagonal 8	(0.1,...,0.1)	1000	Extended PSC1	(3,0.1,....,3,0.1)	4
Diagonal 8	(0.1,...,0.1)	1500	Extended PSC1	(3,0.1,....,3,0.1)	6
Diagonal 8	(0.1,...,0.1)	2000	Extended DENSCHNF	(2,0,...,2,0)	1500
Diagonal 8	(0.1,...,0.1)	2500	Extended DENSCHNF	(2,0,...,2,0)	2000
Extended Himmelblau	(1,...,1)	9000	Extended DENSCHNF	(2,0,...,2,0)	2500
Extended Himmelblau	(1,...,1)	10000	Extended DENSCHNF	(2,0,...,2,0)	3000
FLETCHCR	(0,....,0)	2	Arwhead	(1,...,1)	50
FLETCHCR	(0,....,0)	4	Arwhead	(1,...,1)	70
NONSCOMP	(2,...,2)	2	Arwhead	(1,...,1)	150
Extended DENSCHNB	(2,...,2)	2000	Arwhead	(1,...,1)	200
Extended DENSCHNB	(2,...,2)	3000	HIMMELBG	(0.5,...,0.5)	50000
Extended DENSCHNB	(2,...,2)	5000	HIMMELBG	(0.5,...,0.5)	60000
Extended DENSCHNB	(2,...,2)	6000	LIARWHD	(4,...4)	4000
Generalized Rosenbrok	(0.5,....,0.5)	2	LIARWHD	(4,...4)	5000
Extended Hiebert	(0.1,0,...,0.1,0)	70	LIARWHD	(4,...4)	5500
Extended Hiebert	(0.1,0,...,0.1,0)	500	LIARWHD	(4,...4)	20000
Extended Hiebert	(0.1,0,...,0.1,0)	700	Hager	(1,...,1)	2
Extended Hiebert	(0.1,0,...,0.1,0)	1000	Hager	(1,...,1)	10
Almost Perturbed Quadratic	(0.5,...,0.5)	2	DIXON3DQ	(-1,...,-1)	2

Figures 1-4 show that the curves of the methods CR, ADHCG1 and ADHCG2 dominate the other curves by solving 94% of the test problems successfully, with superiority to the CR method since it is faster then ADHCG1 and ADHCG2 on 78% of the test problems. The PRP and EPF methods have respectively the fourth and fifth best performances with 92% and 91% of test problems, followed by hSM* with 90% of the test problems, whereas RMIL and THCG+ score about 89%. These outcomes demonstrate that the CR method is competitive and converges quickly in the majority of the test problems.

4.1 Image restoration problems

Image restoration is of interest in optimization fields, it aims to recover the original image from an image damaged by impulse noises; its mathematical formulation can be found in [8]. In this subsection, we compare the performance of CR algorithm with the variants studied in the above subsection to solve image restoration problems. In this study, the images of Man.png (512×512), Hill.jpg (512×512), Boat.png (512×512) and Bridge.bmp

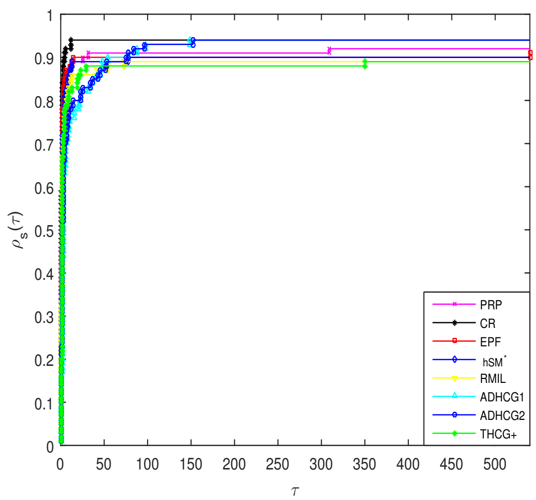

Figure 1: CPU Time performance profile.

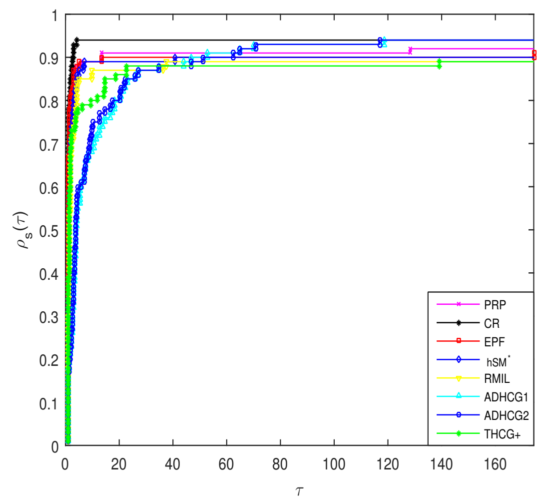

Figure 2: Iterations performance profile.

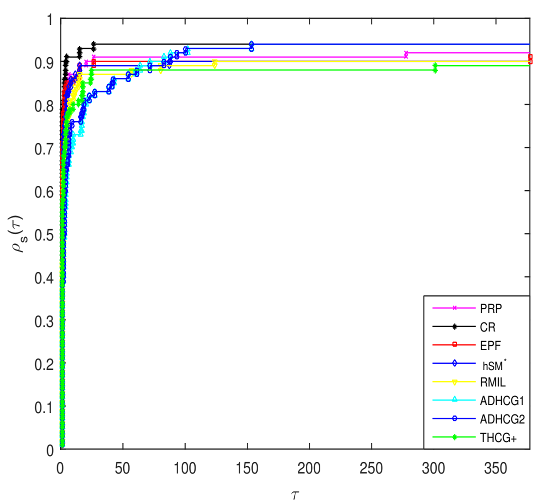

Figure 3: Function evaluations performance profile.

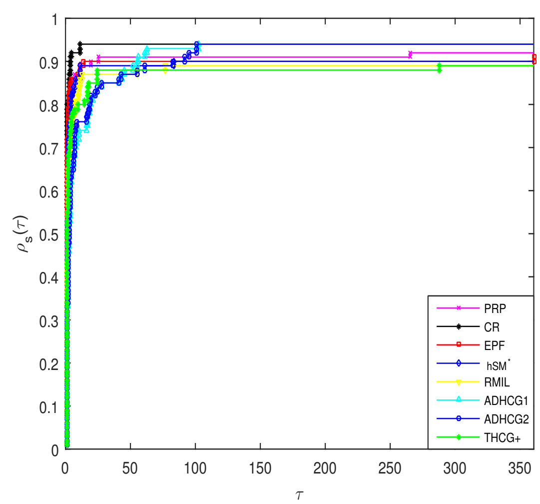

Figure 4: Gradient evaluations performance profile.

(512×512) are selected as test images. The image quality is measured by the parameters: Iter (number of iterations), CPU-time, PSNR (Peak Signal-to-Noise Ratio) and Err (relative error) given by the following formulas:

$$PSNR = 10\log_{10}\frac{M \times N \times 255^2}{\sum_{i,j}(x^r_{i,j} - x^*_{i,j})^2}, \qquad Err = \frac{\|x^r - x^*\|}{\|x^*\|},$$

where $x^r_{i,j}$ and $x^*_{i,j}$ denote respectively the pixel values of the restored image and of the original one, M and N are the sizes of the image. The algorithm that has a large PSNR with small CPU-time and Err is chosen as the best one. The setting parameters are similarly chosen as in the above subsection, and each algorithm will stop as one of

Figure 5: Part I. The noisy images with 30% salt-and-pepper (first row) and the restored images by EPF (second row), RMIL (third row) and PRP (last row).

Figure 5: Part II. The restored images by hSM* (first row), CR (second row), THCGP (third row), ADHCG1 (fourth row) and ADHCG2 (last row).

Figure 6: Part I. The noisy images with 70% salt-and-pepper (first row) and the restored images by EPF (second row), RMIL (third row) and PRP (last row).

Figure 6: Part II. The restored images by hSM* (first row), CR (second row), THCGP (third row), ADHCG1 (fourth row) and ADHCG2 (last row).

the following conditions is fulfilled

$$Iter > 300 \ \text{or} \ \frac{|f(x_{k+1} - f(x_k)|}{|f(x_k)|} < 10^{-4}.$$

The detailed performances for the Man, Hill, Boat and Bridge with 30 % and 70 % of salt-and-pepper noise are illustrated respectively in Figures 5 and 6. The obtained numerical results for the number of iterations, CPU-time, PSNR and the corresponding relative error are displayed in Tables 3 and 2 where the best results are styled in bold.

Methods \ Images		Man	Hill	Boat	Bridge
EPF	Iter	18	16	18	29
	CPU	17.7664	13.0997	17.6929	15.5511
	PSNR	31.5369	34.9496	33.6354	28.5928
	Err	0.0551	0.03722	0.0385	0.0751
RMIL	Iter	15	17	**9**	17
	CPU	21.8868	18.0030	16.6706	21.1122
	PSNR	31.4990	34.8820	32.6893	28.5734
	Err	0.055368	0.0375	0.042920	0.075255
PRP	Iter	**11**	16	11	**14**
	CPU	19.5940	15.9562	19.3073	16.3226
	PSNR	31.3375	34.7344	33.1674	28.5131
	Err	0.056408	0.0381	0.040622	0.075779
hSM*	Iter	15	16	16	15
	CPU	17.6889	12.3968	17.5557	15.3714
	PSNR	31.5501	34.9756	33.6823	28.5813
	Err	0.055043	0.0371	0.038284	0.075186
CR	Iter	17	**15**	17	17
	CPU	**13.8810**	**12.7430**	13.5648	**13.5886**
	PSNR	**31.5597**	34.9693	33.6639	28.5931
	Err	**0.054983**	0.0371	0.038365	0.075084
ADHCG1	Iter	46	43	44	44
	CPU	16.6226	15.9433	17.3495	16.4667
	PSNR	31.5332	34.9288	33.6280	**28.6268**
	Err	0.055151	0.0373	0.038524	**0.074794**
ADHCG2	Iter	46	43	44	44
	CPU	16.5253	16.0104	17.5813	16.4783
	PSNR	31.5332	34.9288	33.6280	**28.6268**
	Err	0.055151	0.0373	0.038524	**0.074794**
THCG+	Iter	29	23	22	25
	CPU	14.5561	13.1561	**13.1498**	13.7842
	PSNR	31.5542	**34.9868**	**33.6844**	28.4667
	Err	0.055017	**0.0370**	**0.038275**	0.076185

Table 2: Numerical results for image restoration problems with 30% salt-and-pepper.

Inspection of Figures 5 and 6 and the results obtained from Tables 3 and 2 shows a satisfactory performance of the CR algorithm. Indeed, it can be seen from the bold values in Table 3 that the proposed algorithm succeeds in restoring the majority of test images with higher PSNR values and overall needs less CPU time.

Methods \ Images		Man	Hill	Boat	Bridge
EPF	Iter	24	29	28	31
	CPU	27.6617	26.4910	34.3126	36.7456
	PSNR	26.2376	29.8219	28.2103	24.5522
	Err	0.1015	0.0671	0.0719	0.1196
RMIL	Iter	23	34	26	**15**
	CPU	51.9098	57.3696	62.9898	51.2787
	PSNR	26.1395	**29.8249**	28.2408	22.4305
	Err	0.102621	**0.0671**	0.071630	0.152643
PRP	Iter	**20**	**16**	**15**	**15**
	CPU	49.1265	28.6253	37.0087	54.1025
	PSNR	26.1119	29.3224	27.5034	22.0518
	Err	0.102948	0.0711	0.077976	0.159447
hSM*	Iter	23	20	18	19
	CPU	34.1524	**22.8480**	32.1868	31.0612
	PSNR	26.2949	29.7088	28.2232	24.4001
	Err	0.100802	0.0680	0.071775	0.121674
CR	Iter	**20**	26	23	16
	CPU	31.6035	32.75028	36.8002	**15.9113**
	PSNR	26.2393	29.7954	28.2483	**28.5640**
	Err	0.101449	0.0673	0.071567	**0.075337**
ADHCG1	Iter	70	64	65	67
	CPU	40.8277	37.3012	37.0822	36.0258
	PSNR	26.1809	29.7161	28.1806	24.3848
	Err	0.102132	0.0679	0.072128	0.121889
ADHCG2	Iter	70	64	65	67
	CPU	40.0976	39.7764	37.0464	36.1044
	PSNR	26.1809	29.7161	28.1806	24.3848
	Err	0.102132	0.0679	0.072128	0.121889
THCG+	Iter	29	28	31	26
	CPU	**26.1769**	27.7290	**26.0725**	24.0349
	PSNR	**26.3218**	29.7459	**28.3121**	24.4231
	Err	**0.100490**	0.0677	**0.071044**	0.121352

Table 3: Numerical results for image restoration problems with 70% salt-and-pepper.

5 Conclusion

In this paper, we have presented a new hybrid nonlinear conjugate gradient method called CR (Combined method for Restoring images); it is a combination of *RMIL* and *hSM* methods. The search direction generated by the CR satisfies the sufficient descent condition independently of any line search and the global convergence is proved under mild conditions. Numerical experiments are carried out on a set of 100 test functions and four image restoration problems with two noise levels. The numerical comparison with some well known classical and recent hybrid CG methods shows that the proposed algorithm is competitive and efficient for solving large-scale complex problems as well as image restoration ones, for our class of problems.

References

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

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

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

[4] R. Fletcher, *Practical Methods of Optimization*, 2nd Ed., J. Wiley, Sons, New York, USA, 1987.

[5] R. Fletcher, C.M. Reeves, *Function minimization by conjugate gradients*, Comput. J. **7** (1964) 149–154.

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

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

[8] Y.I. Ibrahim, H.M. Khudhur, *Modified three-term conjugate gradient algorithm and its applications in image restoration*, J. Electr. Eng. Comput. **28** (2022) 1510–1517.

[9] M. Koontse, P. Kaelo, *Another hybrid conjugate gradient method for unconstrained optimization*, J. Nonlinear Anal. Optim. **2** (2014) 127–137.

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

[11] I.E. Livieris, V. Tampakas, P. Pintelas, *A descent hybrid conjugate gradient method based on the memoryless BFGS update*, Numer. Algor. **79** (2018) 11691185.

[12] M. Lotfi, S.M. Hosseini, *An efficient hybrid conjugate gradient method with sufficient descent property for unconstrained optimization*, Optim. Methods Softw. **37** (2022) 1725–1739.

[13] E. Mehamdia, Y. Chaib, T. Bechouat, *Two modified conjugate gradient methods for unconstrained optimization and applications*, RAIRO-Oper. Res. **57** (2023) 333–350.

[14] P. Mtagulwa, P. Kaelo, *An efficient modified PRP-FR hybrid conjugate gradient method for solving unconstrained optimization problems*, Appl. Numer. Math. **145** (2019) 111–120.

[15] E. Polak, G. Ribiere, *Note sur la convergence de méthodes de directions conjuguées*, Revue Française d'informatique et de Recherche Opérationnelle, Série Rouge **3** (1969) 35–43 .

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

[17] M. Rivaie, M. Mamat, L. W. June, I. Mohd, *A new class of nonlinear conjugate gradient coefficients with global convergence properties*, Appl. Math. Comput. **218** (2012) 11323–11332.

[18] I. M. Sulaiman, N. A. Bakar, M. Mamat, B.A. Hassan, M. Malik, A. M. Ahmed, *A new hybrid conjugate gradient algorithm for optimization models and its application to regression analysis*, J. Electr. Eng. Comput. **23** (2021) 1100–1109.

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

[20] L. Zhang, *An improved Wei-Yao-Liu nonlinear conjugate gradient method for optimization computation*, Appl. Math. Comput. **215** (2009) 2269–2274.

[21] R. Ziadi, R. Ellaia, A. Bencherif-Madani, *Global optimization through a stochastic perturbation of the Polak-Ribiére conjugate gradient method*, J. Comput. Appl. Math. **317** (2017) 672–684.

[22] R. Ziadi, A. Bencherif-Madani, *A mixed algorithm for smooth global optimization*, J. Math. Model. **11** (2023) 207–228.

[23] R. Ziadi, A. Bencherif-Madani, *A Perturbed quasi-Newton algorithm for bound-constrained global optimization*, J. Comp. Math., 2023, doi:10.4208/jcm.2307-m2023-0016.

[24] G. Zoutendijk, *Nonlinear programming computational methods*, J. Integ. Nonlinear Progr. (1970) 37–86.