RAIRO-Oper. Res. 57 (2023) 333–350
https://doi.org/10.1051/ro/2023010

RAIRO Operations Research
www.rairo-ro.org

TWO MODIFIED CONJUGATE GRADIENT METHODS FOR SOLVING UNCONSTRAINED OPTIMIZATION AND APPLICATION

Abd Elhamid Mehamdia[1,2,*], Yacine Chaib[1,2] and Tahar Bechouat[2]

Abstract. Conjugate gradient methods are a popular class of iterative methods for solving linear systems of equations and nonlinear optimization problems as they do not require the storage of any matrices. In order to obtain a theoretically effective and numerically efficient method, two modified conjugate gradient methods (called the MCB1 and MCB2 methods) are proposed. In which the coefficient β_k in the two proposed methods is inspired by the structure of the conjugate gradient parameters in some existing conjugate gradient methods. Under the strong Wolfe line search, the sufficient descent property and global convergence of the MCB1 method are proved. Moreover, the MCB2 method generates a descent direction independently of any line search and produces good convergence properties when the strong Wolfe line search is employed. Preliminary numerical results show that the MCB1 and MCB2 methods are effective and robust in minimizing some unconstrained optimization problems and each of these modifications outperforms the four famous conjugate gradient methods. Furthermore, the proposed algorithms were extended to solve the problem of mode function.

Mathematics Subject Classification. 90C30, 65K05, 62G05.

Received July 27, 2022. Accepted January 27, 2023.

1. Introduction

The optimization model is a requisite mathematical problem since it has been connected to different fields such as economics, engineering and physics. Today there are many optimization algorithms, such as Newton, quasi-Newton and bundle algorithms. Note that these algorithms fail to solve large-scale optimization problems as they need to store and calculate relevant matrices. In contrast, the conjugate gradient (CG) algorithm is successful due to its simplicity of iteration and low memory requirements. In this paper, the nonlinear CG method is studied for the following unconstrained optimization problem:

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

where f is a smooth and nonlinear function. The CG method generates a sequence $\{x_k\}_{k\geq 0}$ such that:

$$x_{k+1} = x_k + \alpha_k d_k, \tag{1.2}$$

Keywords. Unconstrained optimization, conjugate gradient method, numerical comparisons, mode function, kernel estimator.

[1] Laboratory Informatics and Mathematics (LIM), Mohamed Cherif Messaadia University, Souk Ahras 41000, Algeria.

[2] Mohamed Cherif Messaadia University, Souk Ahras 41000, Algeria.

[*]Corresponding author: a.mehamdia@univ-soukahras.dz

where x_k is the current iteration point and $d_k \in \mathbb{R}^n$ is the search direction defined by the following formula:

$$d_{k+1} = -g_{k+1} + \beta_k d_k; \; d_0 = -g_0, \tag{1.3}$$

where g_{k+1} the gradient of f at x_{k+1} and the parameter β_k is known as the conjugate gradient coefficient. The step length α_k is very important for the global convergence of conjugate gradient methods. One often requires the line search to satisfy the Wolfe conditions (WLS)

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

and

$$g_{k+1}^T d_k \geq \sigma g_k^T d_k. \tag{1.5}$$

Also, the strong Wolfe (SWLS) conditions consist of (1.4) and

$$\left| g_{k+1}^T d_k \right| \leq -\sigma g_k^T d_k, \tag{1.6}$$

where $0 < \delta < \sigma < 1$. For the scalar β_k many formulas have been proposed. Some of the classical algorithms for β_k are Fletcher–Reeves (FR) method [12], Dai–Yuan (DY) method [5], Conjugate Descent (CD) method proposed by Fletcher [11], Polak–Ribière and Polyak (PRP) method [20,21], Hestenes-Stiefel (HS) method [13] and Liu-Storey (LS) method [17]. where formulas for β_k, are given, respectively, by:

$$\beta_k^{\text{FR}} = \frac{\|g_{k+1}\|^2}{\|g_k\|^2}, \; \beta_k^{\text{DY}} = \frac{\|g_{k+1}\|^2}{y_k^T d_k}, \; \beta_k^{\text{CD}} = \frac{\|g_{k+1}\|^2}{-g_k^T d_k}, \tag{1.7}$$

$$\beta_k^{\text{PRP}} = \frac{g_{k+1}^T y_k}{\|g_k\|^2}, \quad \beta_k^{\text{HS}} = \frac{g_{k+1}^T y_k}{y_k^T d_k}, \quad \beta_k^{\text{LS}} = \frac{g_{k+1}^T y_k}{-g_k^T d_k}, \tag{1.8}$$

where $\|\cdot\|$ denotes the Euclidean norm and $y_k = g_{k+1} - g_k$. Although all nonlinear conjugate gradient methods should reduce to the linear conjugate gradient method when f is a convex quadratic and the line search is exact, their convergence properties may be quite different for nonquadratic functions. For example, Al-Baali [1] established the convergence of the FR method if the step length α_k satisfies (1.4) and (1.6) with $\sigma < \frac{1}{2}$. Dai and Yuan [5] proved the global convergence result of the DY if the WLS is used with $\sigma < 1$. In contrast, the PRP and HS methods have a drawback in that they may not globally be convergent even with the exact line search [22]. This problem inspired numerous researchers to study the global convergence of the above methods under the inexact line search. Wei *et al.* [25] based on the PRP method, gave a new CG formula, called WYL method, where β_k is given by:

$$\beta_k^{\text{WYL}} = \frac{g_{k+1}^T \left(g_{k+1} - \frac{\|g_{k+1}\|}{\|g_k\|} g_k \right)}{\|g_k\|^2}.$$

The WYL method can be considered as a modification of the PRP method, which inherits good numerical results from the original method. Furthermore, Huang *et al.* [14] proved that the WYL method satisfies the sufficient descent condition and converges globally under the SWLS with the parameter $\sigma < \frac{1}{4}$. Moreover, the Wei–Yao–Liu method may not be a descent method if the WLS is used. Yao *et al.* [26] extended this idea to the HS method, called the MHS. The parameter β_k in the MHS method is given by:

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

For the SWLS under the Lipschitz continuity of the gradient, Yao *et al.* [26] established the global convergence of this computational scheme. Soon afterward, Zhang [27] took a little modification to the WYL method and constructed the NPRP method, such as the CG coefficient is computed by:

$$\beta_k^{\text{NPRP}} = \frac{\|g_{k+1}\|^2 - \frac{\|g_{k+1}\|}{\|g_k\|} \left| g_{k+1}^T g_k \right|}{\|g_k\|^2}.$$

Also, this author [27] has given a modified HS method, in which β_k is defined by:

$$\beta_k^{\mathrm{NHS}} = \frac{\|g_{k+1}\|^2 - \frac{\|g_{k+1}\|}{\|g_k\|}\left|g_{k+1}^T g_k\right|}{y_k^T d_k}.$$

The NPRP and NHS methods possess sufficient descent conditions and converge globally if the SWLS is used and the parameter σ is restricted in $\left(0, \frac{1}{2}\right)$. Numerical results reported in [27] show that the NPRP method performs better than the WYL method and the NHS method better than the MHS method. Likewise, Du *et al.* [9] in 2016 give two modified CG methods, proposing the following formula:

$$\beta_k^{\mathrm{NVPRP}^*} = \frac{g_{k+1}^T\left(g_{k+1} - \frac{\left|g_{k+1}^T g_k\right|}{\|g_k\|^2} g_k\right)}{\|g_k\|^2},$$

and

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

The NVPRP* and NVHS* methods have sufficient descent conditions and are globally convergent if the SWLS is utilized with the parameter $\sigma < \frac{1}{4}$ and $\sigma < \frac{1}{3}$, respectively. In 2012, Dai and Wen [6] proposed two modified CG methods, denoted by DPRP and DHS methods. The parameter β_k in the DPRP method is given by:

$$\beta_k^{\mathrm{DPRP}} = \frac{\|g_{k+1}\|^2 - \frac{\|g_{k+1}\|}{\|g_k\|}\left|g_{k+1}^T g_k\right|}{\|g_k\|^2 + \mu\left|d_k^T g_{k+1}\right|},\ \mu > 1.$$

And the scalar β_k in the DHS method is defined as:

$$\beta_k^{\mathrm{DHS}} = \frac{\|g_{k+1}\|^2 - \frac{\|g_{k+1}\|}{\|g_k\|}\left|g_{k+1}^T g_k\right|}{y_k^T d_k + \mu\left|d_k^T g_{k+1}\right|}, \mu > 1.$$

The convergence of two methods with the WLS established and numerical results show that these computational schemes are efficient [6]. Recently, Zhu *et al.* [28] gave a modified CG method, called DDY1, where β_k in this method given by:

$$\beta_k^{\mathrm{DDY1}} = \begin{cases} \dfrac{\|g_{k+1}\|^2 - \frac{\mu_1\left(g_{k+1}^T d_k\right)^2 g_{k+1}^T g_k}{\|g_k\|\|g_{k+1}\|\|d_k\|^2}}{y_k^T d_k}, & g_{k+1}^T g_k \geq 0, \\ \dfrac{\|g_{k+1}\|^2 + \frac{\mu_1\left(g_{k+1}^T d_k\right)^2 g_{k+1}^T g_k}{\|g_k\|\|g_{k+1}\|\|d_k\|^2}}{y_k^T d_k}, & g_{k+1}^T g_k < 0, \end{cases}$$

with $\mu_1 \in [0, 1]$. The authors proved that this method possesses sufficient descent conditions and global convergence property when SWLS is employed [28].

The aim of this paper is to propose two new conjugate gradient methods. Under the SWLS, the convergence properties of MCB1 and MCB2 CG methods are established. Numerical results show that the two modifications are efficient and robust. Finally, an application of these methods in nonparametric mode estimator is also considered.

The most important and new thing in this work is the application of these methods in nonparametric statistics, where we are the first to use and access in this field.

– This work is organized as follows. In the next section, the two modified algorithms are introduced and the sufficient descent condition is proved. In setion three, the global convergence of the two proposed methods with an SWLS is proved. The numerical results are contained in section four. In section five, an application of the new methods in statistics nonparametric is focused. Finally, a paper summary is made.

2. New conjugate gradient methods

In this section, The two novel β_k are introduced, which are defined as β_k^{MCB1} and β_k^{MCB2}, where MCB denoted Mehamdia, Chaib and Bechouat. First, the conjugate gradient parameter of MCB1 method is presented as follows

$$\beta_k^{\text{MCB1}} = \frac{\|g_{k+1}\|^2 - \rho_1 |g_{k+1}^T g_k| \omega_k}{d_k^T y_k + \mu_1 |d_k^T g_{k+1}|}, \tag{2.1}$$

where $\omega_k = \frac{|g_{k+1}^T d_k| g_{k+1}^T d_k}{\|g_k\| \|g_{k+1}\| \|d_k\|^2}$, $\rho_1 \in [0,1]$ and $\mu_1 > 1 + \rho_1$.

Second, the parameter β_k of the MCB2 method is defined as follows

$$\beta_k^{\text{MCB2}} = \frac{\|g_{k+1}\|^2 - \rho_2 |g_{k+1}^T g_k| \omega_k}{\|g_k\|^2 + \mu_2 |d_k^T g_{k+1}|}, \tag{2.2}$$

where $\rho_2 \in [0,1]$ and $\mu_2 > 1 + \rho_2$.

The search direction d_k of MCB1 algorithm is given by:

$$d_{k+1} = -g_{k+1} + \beta_k^{\text{MCB1}} d_k;\ d_0 = -g_0, \tag{2.3}$$

and the search direction d_k of MCB2 algorithm is defined by:

$$d_{k+1} = -g_{k+1} + \beta_k^{\text{MCB2}} d_k;\ d_0 = -g_0. \tag{2.4}$$

2.1. Algorithms

In this part, the MCB1 and MCB2 Algorithms with the SWLS are presented.

MCB1 Algorithm

Step 1. Initialization.
Choose an initial point $x_0 \in \mathbb{R}^n$ and the parameters $0 < \delta < \sigma < 1$. Compute $f(x_0)$ and g_0. Set $d_0 = -g_0$.
Step 2. Test for a continuation of iterations.
If $\|g_k\|_\infty \leq 10^{-6}$, then stop. Otherwise, go to the next step.
Step 3. Line search.
Calculate α_k satisfies the linear search conditions of stong Wolfe (1.4) and (1.6) and update the variables $x_{k+1} = x_k + \alpha_k d_k$.
Step 4. Compute β_k by the formula (2.1).
Step 5. Compute the search direction by the formula (2.3).
Step 6. Set $k = k + 1$ and go to Step 2.

MCB2 Algorithm

The Algorithm of MCB2 is the same as the MCB1 Algorithm, but in Step 4, we replace formula (2.1) by formula (2.2) and in Step 5, we replace equation (2.3) by equation (2.4).

2.2. The sufficient descent direction

– The following lemma to prove the sufficient descent direction of proposed methods is needed.

Lemma 2.1. *The following inequality always holds:*

$$\|g_{k+1}\|^2 - \rho |g_{k+1}^T g_k| \omega_k \leq (1+\rho) \|g_{k+1}\|^2, \text{ with } \rho \in [0,1]. \tag{2.5}$$

Proof. Suppose that the ξ_k is the angle between the g_k and g_{k+1} vectors and the θ_k is the angle between the g_{k+1} and d_k vectors, then

$$\cos\xi_k = \frac{g_{k+1}^T g_k}{\|g_{k+1}\|\|g_k\|}, \ \cos\theta_k = \frac{g_{k+1}^T d_k}{\|g_{k+1}\|\|d_k\|}.$$

We have

$$\begin{aligned}\|g_{k+1}\|^2 - \rho\left|g_{k+1}^T g_k\right|\omega_k &= \|g_{k+1}\|^2(1-\rho|\cos\theta_k||\cos\xi_k|\cos\theta_k)\\ &\le \|g_{k+1}\|^2\left(1+\rho\cos^2\theta_k|\cos\xi_k|\right)\\ &\le (1+\rho)\|g_{k+1}\|^2.\end{aligned}$$

The result can be achieved. □

– First, the sufficient descent direction of the MCB1 is proved.

Theorem 2.1. *Let the sequence* $\{g_k\}_{k\ge0}$ *and* $\{d_k\}_{k\ge0}$ *be generated by Algorithm MCB1, then for positive constant* c_1 *we have*

$$g_k^T d_k \le -c_1\|g_k\|^2, \qquad \forall k \geqslant 0. \tag{2.6}$$

Proof. The following proof is by induction. For $k=0$, $g_0^T d_0 = -\|g_0\|^2$, thus the sufficient descent condition holds for $k=0$. Now, it is assumed that (2.6) holds for k and prove that for $k+1$.

Multiplying (2.3) by g_{k+1}^T from the left, we get

$$g_{k+1}^T d_{k+1} = -\|g_{k+1}\|^2 + \frac{\|g_{k+1}\|^2 - \rho_1\left|g_{k+1}^T g_k\right|\omega_k}{d_k^T y_k + \mu_1\left|d_k^T g_{k+1}\right|} g_{k+1}^T d_k. \tag{2.7}$$

From (1.6) and (2.6), it is obtained

$$d_k^T y_k \ge (1-\sigma)\left(-d_k^T g_k\right) \ge 0. \tag{2.8}$$

By (2.5), (2.7) and (2.8), it is got

$$g_{k+1}^T d_{k+1} \le -\|g_{k+1}\|^2 + \frac{(1+\rho_1)\|g_{k+1}\|^2\left|g_{k+1}^T d_k\right|}{\mu_1\left|d_k^T g_{k+1}\right|}. \tag{2.9}$$

Hence,

$$g_{k+1}^T d_{k+1} \le -c_1\|g_{k+1}\|^2, \tag{2.10}$$

where $c_1 = 1 - \frac{1+\rho_1}{\mu_1}$, so there is a constant $c_1 > 0$ with $\mu_1 > 1+\rho_1$. Therefore, the proof is completed. □

– Second, the sufficient descent direction of the MCB2 method is proved.

Theorem 2.2. *Let the direction* d_k *be yielded by the MCB2 Algorithm. Then, we get*

$$g_k^T d_k \le -c_2\|g_k\|^2, \qquad \forall k \geqslant 0, \tag{2.11}$$

where $c_2 = 1 - \frac{1+\rho_2}{\mu_2}$.

Proof. The following proof is by induction. For $k=0$, $g_0^T d_0 = -\|g_0\|^2$, thus the sufficient descent condition holds for $k=0$. Now, it is assumed that (2.11) holds for k and prove that for $k+1$.

Multiplying (2.4) by g_{k+1}^T from the left, we obtain

$$g_{k+1}^T d_{k+1} = -\|g_{k+1}\|^2 + \frac{\|g_{k+1}\|^2 - \rho_2\left|g_{k+1}^T g_k\right|\omega_k}{\|g_k\|^2 + \mu_2\left|d_k^T g_{k+1}\right|} g_{k+1}^T d_k. \tag{2.12}$$

From the (2.5) and (2.12), there is

$$g_{k+1}^T d_{k+1} \leq -\|g_{k+1}\|^2 + \frac{(1+\rho_2)\|g_{k+1}\|^2}{\mu_2 |g_{k+1}^T d_k|} |g_{k+1}^T d_k|. \tag{2.13}$$

Thus,

$$g_{k+1}^T d_{k+1} \leq -c_2 \|g_{k+1}\|^2.$$

So there is a constant $c_2 > 0$ with $\mu_2 > 1 + \rho_2$. Hence, Theorem 2.2 is proved. □

3. Global convergence

To establish the global convergence of proposed methods, the following basic Assumptions on the objective function are needed.

Assumption 3.1. *The level set*

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

is bounded.

Assumption 3.2. *In some open convex neighborhood $\mathcal{N}$ of S, the function f is continuously differentiable and its gradient is Lipschitz continuous, namely, there exists a constant $L > 0$, such that*

$$\|\nabla f(x) - \nabla f(y)\| \leq L\|x - y\|, \qquad \forall x, y \in \mathcal{N}. \tag{3.1}$$

From Assumption 3.2, it is deduced that for all $x \in \mathcal{N}$, there exists a positive constant $\Gamma \geq 0$, such that

$$\| \nabla f(x) \| \leq \Gamma, \quad \text{for all } x \in \mathcal{N}. \tag{3.2}$$

– It follows from Dai *et al.* [7] proved the sufficient condition for the convergence of CG methods with strong Wolfe line search.

Lemma 3.1. *Let Assumptions 3.1 and 3.2 hold. Consider the method (1.2) and (1.3), where d_k is a descent direction and α_k is obtained by the SWLS. If*

$$\sum_{k \geq 0} \frac{1}{\|d_k\|^2} = \infty,$$

then

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

– This lemma is also needed to prove the convergence of MCB1 and MCB2 methods.

Lemma 3.2. *Let Assumptions 3.1 and 3.2 hold. If d_k is a descent direction and α_k satisfies the WLS (1.4) and (1.5). Then*

$$\alpha_k \geq \frac{(1-\sigma) \mid g_k^T d_k \mid}{L \parallel d_k \parallel^2}. \tag{3.3}$$

Proof. See the proof of Lemma 3.2 in Lui and Li [16]. □

Remark 3.1. From (1.6), (2.6) and (2.11), the step size α_k obtained in the MCB1 and MCB2 Algorithms satisfies (3.3). This indicates the step length α_k obtained in the MCB1 and MCB2 methods is not equal to zero, *i.e.*, there exists a constant $\lambda > 0$, such that

$$\alpha_k \geq \lambda, \qquad \forall k \geqslant 0. \tag{3.4}$$

– The following theorem establishes the global convergence of the MCB1 method with the SWLS.

Theorem 3.1. *Suppose that Assumptions 3.1 and 3.2 hold. Consider any CG method in the form (1.2) and (1.3), with the parameter $\beta_k = \beta_k^{\mathrm{MCB1}}$, in which the step length α_k is determined to satisfy the SWLS condition (1.4) and (1.6), where d_k is a descent search direction. Then, this method converges in the sense that*

$$\lim_{k\to\infty} \inf\|g_k\| = 0. \tag{3.5}$$

Proof. To prove Theorem 3.1, contradiction is used. That is, if equation (3.5) is not true, then we can find a positive constant γ_1, such that

$$\|g_k\| \geq \gamma_1, \qquad \text{for all } k. \tag{3.6}$$

Since the definition of β_k^{MCB1} and (2.5) is

$$\left|\beta_k^{\mathrm{MCB1}}\right| \leq \frac{(1+\rho_1)\|g_{k+1}\|^2}{d_k^T y_k + \mu_1\left|d_k^T g_{k+1}\right|} \leq \frac{(1+\rho_1)\|g_{k+1}\|^2}{d_k^T y_k}. \tag{3.7}$$

By using (2.6), (2.8) and (3.6),

$$d_k^T y_k \geq (1-\sigma)\left(-d_k^T g_k\right) \geq c_1(1-\sigma)\gamma_1^2 \geq 0. \tag{3.8}$$

Hence, by (3.2), (3.7) and (3.8),

$$\left|\beta_k^{\mathrm{MCB1}}\right| \leq \frac{(1+\rho_1)\Gamma^2}{c_1(1-\sigma)\gamma_1^2} = E. \tag{3.9}$$

Thus, it follows from (2.3), (3.2), (3.4) and (3.9) that

$$\|d_{k+1}\| \leq \| g_{k+1} \| + \left|\beta_k^{\mathrm{MCB1}}\right| \frac{\| x_{k+1} - x_k \|}{\alpha_k} \leq M_1, \tag{3.10}$$

where

$$M_1 = \Gamma + E\frac{D}{\lambda},$$

and

$$D = \max\{\|y - z\| : y, z \in \mathscr{N}\}.$$

By taking the summation $k \geq 0$,

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

By applying Lemma 3.1, equation (3.5) is true. This is a contradiction with (3.6), so we have proved (3.5). □

– The next lemma, called the Zoutendijk condition, is often used to prove the global convergence of MCB2 CG method. It was originally given by Zoutendijk [29].

Lemma 3.3. *It is assumed that x_0 is a starting point for which Assumptions 3.1 and 3.2 hold. Consider any method in the form (1.2) and (1.3), where d_k is a descent direction and the step size α_k satisfies the WLS (1.4) and (1.5), then we have*

$$\sum_{k=0}^{\infty} \frac{(g_k^T d_k)^2}{\|d_k\|^2} < \infty. \tag{3.11}$$

It is easy to get from (2.11) that the Zoutendijk condition (3.11) is equivalent to the following inequality

$$\sum_{k=0}^{\infty} \frac{\|g_k\|^4}{\|d_k\|^2} < \infty. \tag{3.12}$$

– The following theorem is used to prove the global convergence of the MCB2 method.

Theorem 3.2. *Consider that Assumptions 3.1 and 3.2 hold. Let the sequences* $\{g_k\}_{k\geq 0}$ *and* $\{d_k\}_{k\geq 0}$ *be generated by MCB2 Algorithm. Then*

$$\lim_{k\to\infty} \inf \|g_k\| = 0. \tag{3.13}$$

Proof. Suppose that (3.13) does not hold. Then, there exists a constant $\gamma_2 > 0$ such that

$$\|g_k\| \geq \gamma_2, \qquad \text{for all } k. \tag{3.14}$$

In fact, by using (2.2) and (2.5),

$$\left|\beta_k^{\mathrm{MCB2}}\right| \leq \frac{(1+\rho_2)\|g_{k+1}\|^2}{\|g_k\|^2 + \mu_2\left|d_k^T g_{k+1}\right|} \leq \frac{(1+\rho_2)\|g_{k+1}\|^2}{\|g_k\|^2}. \tag{3.15}$$

Therefore, from (3.2), (3.14) and (3.15),

$$\left|\beta_k^{\mathrm{MCB2}}\right| \leq \frac{(1+\rho_2)\Gamma^2}{\gamma_2^2} = F. \tag{3.16}$$

Thus, it follows from (2.4), (3.2), (3.4) and (3.16) that

$$\|d_{k+1}\| \leq \| g_{k+1} \| + \left|\beta_k^{\mathrm{MCB2}}\right| \frac{\| x_{k+1} - x_k \|}{\alpha_k} \leq M_2, \tag{3.17}$$

where

$$M_2 = \Gamma + F\frac{D}{\lambda},$$

and

$$D = \max\{\|y - z\| : y, z \in \mathscr{N}\}.$$

Which implies that

$$\sum_{k\geq 0} \frac{1}{\|d_k\|^2} = \infty. \tag{3.18}$$

On the other hand, from (3.12) and (3.14),

$$\gamma_2^4 \sum_{k\geq 0} \frac{1}{\|d_k\|^2} \leq \sum_{k\geq 0} \frac{\|g_k\|^4}{\|d_k\|^2} < \infty.$$

Which contradicts with (3.18). Hence, equation (3.14) does not hold and the claim (3.13) is proved. □

4. Numerical experiments

In this section, some obtained numerical experiments are presented with the two new proposed CG methods. The test problems have been taken to the CUTE library [2,4]. All the algorithms have been coded in MATLAB 2013 and compiler settings on the PC machine (2.5 GHz, 3.8 GB RAM) with Windows XP operating system. The computational results of MCB1 method are compared with the NHS [27], NVHS* [9], DDY1 [28] and DHS [6]. On the other hand, the computational results of the MCB2 method are compared with the NPRP [27], NVPRP* [9], DDY1 [28] and DPRP [6]. In this numerical result, all algorithms implement the SWLS condition with $\delta = 10^{-3}$ and $\sigma = 10^{-1}$. The iteration is terminated if one of the following conditions is satisfied (i) $\|g_k\|_\infty < 10^{-6}$, where $\|\cdot\|_\infty$ is the maximum absolute component of a vector, (ii) The number of iterations

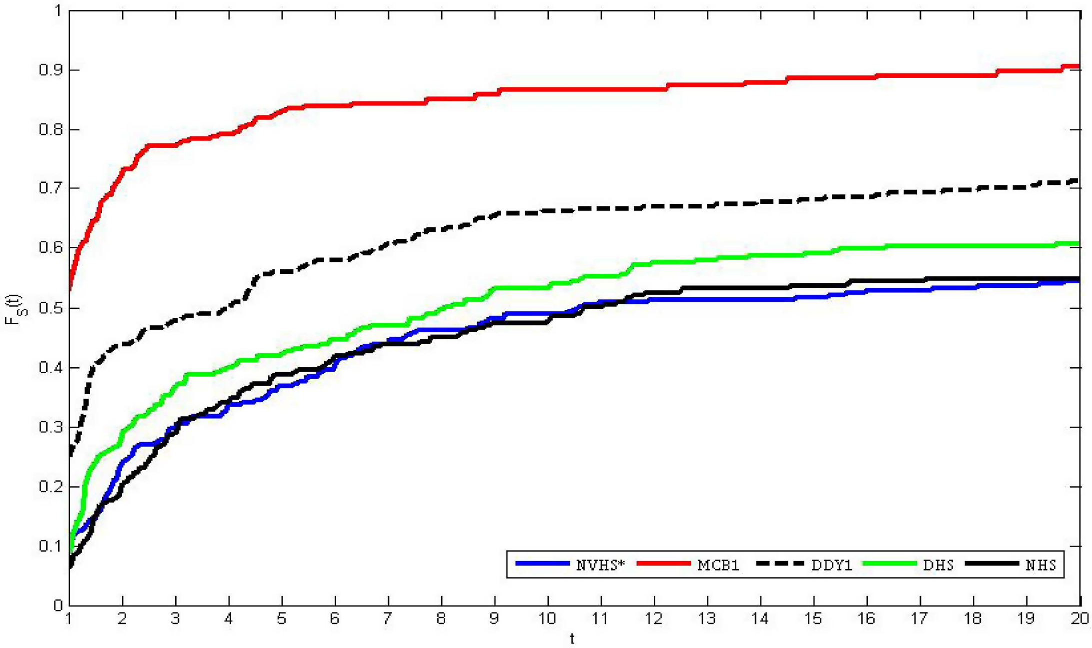

FIGURE 1. Performance profile on the CPU time (MCB1).

exceeded 2000, (iii) The computing time is more than 500 s. The performance profile introduced by Dolan and Morè [8] is chosen to compare the performance according to the number of iterations and CPU time to rule as follows. Let S is the set of methods and P is the set of the test problems with n_p, n_s is the number of the test problems and the number of the methods, respectively. For each problem $p \in P$ and solver $s \in S$, denote $\tau_{p,s}$ be the number of iterations or CPU time required to solve problems $p \in P$ by solver $s \in S$. Then a comparison between different solvers based on the performance ratio is given by

$$r_{p,s} = \frac{\tau_{p,s}}{\min\{\tau_{p,i}, \quad 1 \leq i \leq n_s\}}.$$

Suppose that a parameter $r_M \geq r_{p,s}$ for all problems and solvers chosen, and $r_M = r_{p,s}$ if and only if solver S does not solve problem p. The overall evaluation of the performance of the solvers is then given by the performance profile function given by

$$F_s(t) = \frac{\text{size}\{p : 1 \leq p \leq n_p,\ r_{p,s} \leq t\}}{n_p},$$

where $t \geq 1$ and size$\{p : 1 \leq p \leq n_p,\ r_{p,s} \leq t\}$ is the number of elements in the set $\{p : 1 \leq p \leq n_p,\ r_{p,s} \leq t\}$. This function $F_s : [1, \infty[\to [0, 1]$ is the distribution function for the performance ratio. The value of $F_s(1)$ is the probability that the solver will win the rest of the solvers.

In this numerical study, Dim denotes the dimension of the problem, ITR denotes the number of iterations, TIME denotes the CPU time and Inf denotes the algorithm failed to yield a solution for the problem.

Figure 1 gives a performance comparison of the MCB1 method *versus* NHS, NVHS*, DDY1 and DHS methods. As this figure indicates, the new algorithm prevailed over all other Methods, with respect to CPU time, this clearly confirms the effectiveness of the MCB1 method. Generally, the DDY1 method and the DHS method are better than the NHS and NVHS* methods.

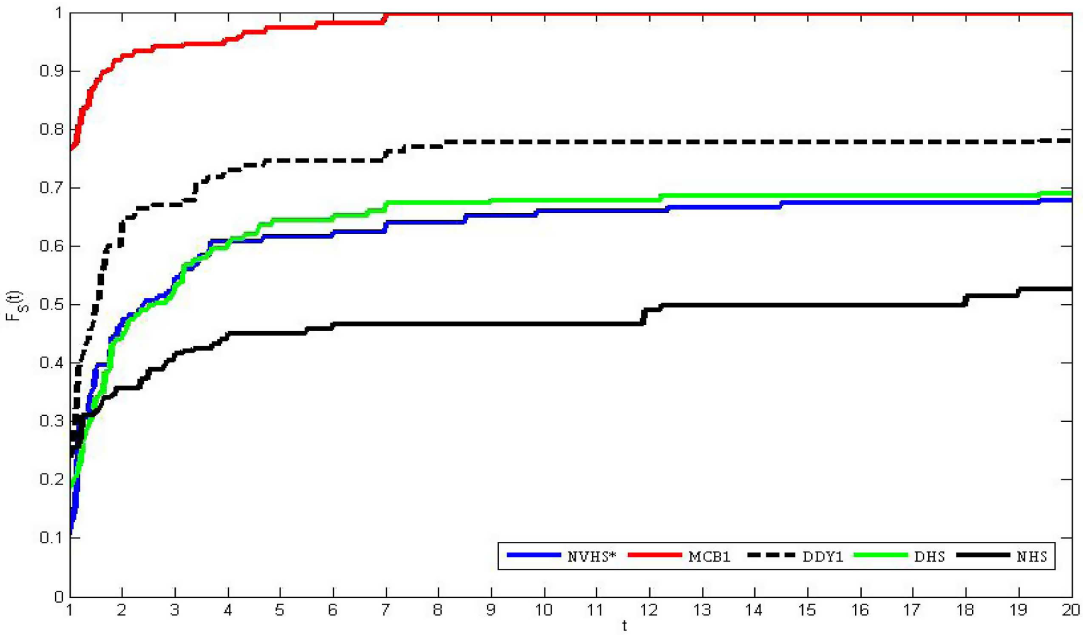

FIGURE 2. Performance profile on the number of iterations (MCB1).

It can be seen from Figure 2 that the MCB1 curve is mostly at the top of the NHS, NVHS*, DDY1 and DHS CG curves, so it is indicating that the MCB1 algorithm outperforms the NHS, NVHS*, DDY1 and DHS methods based on the number of iterations. In particular, the DHS method outperforms the other methods except for the DDY1 method.

From Table 1, it is clear that the average performance of the MCB1, NHS, NVHS*, DDY1 and DHS methods are very similar to the results obtained from Figures 1 and 2.

On the other side, Figure 3 is the performance profile of the MCB2, NPRP, NVPRP*, DDY1 and DPRP CG methods. From this figure, it is concluded that the MCB2 method performs better than the NPRP, NVPRP*, DDY1 and DPRP CG methods, from the viewpoint of the CPU time. Furthermore, although Figure 3 shows that DPRP method is also faster and more robust than DDY1 method when $1.5 < t < 3.5$. Generally DDY1 is preferable to DPRP, NVPRP* and NPRP methods. The NVPRP* method behaves like the NPRP method, for the given test problems.

Figure 4 shows the performance profile for the number of iterations. Relative to this metric, MCB2 achieves the top performance, followed by DDY1 if $t \geq 9$, then DPRP. The NVPRP* method behaves such as the NPRP method.

From Table 2, it is clear that the average performance of the MCB2, NPRP, NVPRP*, DDY1 and DPRP methods are very similar to the results obtained from Figures 3 and 4.

5. Application in mode function

The conjugate gradient method has played an important role in solving large scale unconstrained optimization problems that may arise in regression analysis [30], portfolio selection [3] and image restoration problems [18].

Estimation nonparametric has received a great deal of attention in both theoretical and applied statistics literature. For the historical and mathematical survey, we refer to the reader to Sager [23]. In statistics, it is always interesting to study the central tendency of the data, that is usually quantified using the location

TABLE 1. The simulation results of MCB1, DDY1, NVHS *, DHS and NHS methods.

Method / Function	Dim	MCB1 TIME	MCB1 ITR	NVHS* TIME	NVHS* ITR	NHS TIME	NHS ITR	DDY1 TIME	DDY1 ITR	DHS TIME	DHS ITR
Griewank	2000	0.1410	7	5.1410	140	72.0500	1213	5.5890	140	6.1620	140
	2500	0.1840	7	8.0000	225	25.5610	338	8.0790	225	8.0310	225
	3000	0.1100	7	13.5780	292	14.3590	175	16.2790	291	15.9070	291
Dixon	5000	0.4850	4	1.0460	7	0.9220	7	0.9210	6	1.0000	7
	6000	0.7030	4	1.1870	7	0.9380	6	0.9690	6	1.2660	7
	7000	0.7810	4	1.7500	7	1.5470	6	1.1870	6	1.3600	7
Diagonal 4	4000	0.2500	8	0.2810	8	0.2660	8	0.2970	8	0.2660	8
	5000	0.2500	8	0.2970	8	0.2500	8	0.2650	8	0.2340	8
	7000	0.3440	8	0.3750	8	0.3280	8	0.4370	8	0.3590	8
Diagonal 1	600	9.5780	526	18.7180	837	Inf	Inf	Inf	Inf	Inf	Inf
Matyas	2	0.0320	4	0.0780	10	0.0160	4	0.0160	4	0.0160	4
Staircase S3	70	0.2500	226	0.2960	270	0.8910	835	0.4060	385	0.9690	902
	100	0.7660	354	0.5630	387	2.7350	1362	0.8750	597	3.7960	1623
	130	0.6700	424	1.0310	527	1.5310	1895	2.0580	888	3.3430	1244
Arwhead	2000	0.2980	5	0.0940	2	0.7580	10	0.0940	2	0.1090	2
	4000	0.5780	5	0.1720	2	1.2820	10	0.1720	2	0.1720	2
	10 000	1.4220	5	0.3910	2	3.2810	10	0.4220	2	0.4690	2
Quartcim	1100	0.3360	5	14.5620	1102	13.9530	1058	0.3750	8	14.2330	1081
	1200	0.2650	5	18.3890	1136	17.5970	1083	0.4910	8	18.1660	1115
	1800	0.3910	5	0.8430	10	26.2970	1290	1.0980	7	29.4520	1289
	2000	0.4530	5	0.7650	8	30.0150	1332	0.8150	7	34.0750	1335
Linear Perturbed	600	2.7180	262	3.5300	311	9.8300	703	Inf	Inf	11.4060	809
	621	2.7810	251	3.5310	311	8.7970	657	Inf	Inf	8.1880	630
	650	3.2500	273	3.6570	314	13.1720	314	Inf	Inf	9.0780	649
	800	4.3280	298	6.2570	340	25.9660	342	Inf	Inf	23.6710	1086
Extended Hiebert	1200	17.7640	203	Inf	Inf	Inf	Inf	18.7030	328	Inf	Inf
Leon	2	0.2360	34	0.2350	76	0.7030	34	0.2350	61	0.4220	67
Extended White and Holst	5000	1.1560	5	1.7970	7	1.1870	5	1.1620	5	1.1650	5
	5500	1.2660	5	1.8590	7	12560	5	1.4220	5	1.4530	5
	6000	1.6250	5	1.8600	6	1.4840	5	2.0560	5	1.4380	5
Branin	2	0.0160	4	0.0310	8	0.1250	16	0.0620	8	0.0470	8
Cube	1670	0.3910	4	Inf	Inf	0.4380	5	Inf	Inf	Inf	Inf
Almost Perturbed Quartic	1500	0.5000	6	0.5080	7	0.5470	7	0.5000	6	0.5080	7
	5000	1.4370	6	1.9380	7	1.6410	6	1.6720	6	1.6250	6
Prod 2	1800	0.2030	5	0.5310	6	0.2340	6	0.2190	8	0.2350	6
Harkerp	1400	0.4370	9	6.6560	111	6.4220	110	1.2340	23	6.3910	110
Booth	2	0.0310	4	0.0470	8	0.2040	22	0.0470	8	0.1560	21
Quadratic	1000	3.2720	162	3.5710	200	Inf	Inf	3.3430	204	8.2500	424
	1200	4.1290	227	5.8120	269	Inf	Inf	4.8590	244	23.9770	880
	1500	6.9200	254	6.6770	269	Inf	Inf	7.4220	273	36.3990	1033
Quartic	1600	0.1400	5	2.0940	114	17.3900	201	2.0780	114	2.0780	114
	1700	0.1620	5	1.8910	97	19.2970	207	1.6290	97	1.8750	97
	2000	0.1800	5	3.1320	125	46.4100	316	3.1090	124	3.0940	125
	2400	0.1880	5	3.1870	114	49.1510	329	3.6300	114	3.1580	114
Ridge	1200	0.3910	56	Inf	Inf	Inf	Inf	0.5470	78	Inf	Inf
	1800	0.6850	58	Inf	Inf	Inf	Inf	0.5160	57	Inf	Inf
Raydan 2	100	0.1400	6	0.1400	93	4.3450	509	Inf	Inf	0.1880	126
	140	0.1410	41	Inf	Inf	Inf	Inf	0.1720	46	Inf	Inf
Styblinski	800	0.7190	17	1.3750	31	0.9280	18	1.3440	30	1.3880	30
	1960	1.0150	21	3.3210	66	1.4200	25	1.3130	30	2.7960	56

TABLE 1. continued.

Method / Function	Dim	MCB1 TIME	MCB1 ITR	NVHS* TIME	NVHS* ITR	NHS TIME	NHS ITR	DDY1 TIME	DDY1 ITR	DHS TIME	DHS ITR
Sumsquares	1200	2.0820	62	2.0940	62	5.8600	97	7.1410	80	2.0000	61
	1250	2.0000	59	2.1720	63	Inf	Inf	14.7930	120	2.7350	76
	1300	2.8220	64	3.1850	65	Inf	Inf	2.9680	66	2.4540	62
	1600	3.2760	71	3.3130	74	Inf	Inf	3.5780	79	3.7810	84
	1800	3.8280	78	3.9070	78	Inf	Inf	3.9220	78	4.6270	88
Sphere	4200	0.1610	5	0.4470	9	0.7660	9	1.9220	17	0.2820	9
	5400	0.2030	5	0.3540	9	1.0780	10	2.4530	17	0.3750	9
	6500	0.2660	5	0.4060	9	1.3590	10	3.0620	17	0.4530	9
	7800	0.2970	5	0.5470	9	1.3910	9	3.6250	17	0.5310	9
Schwefel 223	7000	0.3910	2	0.4690	3	0.7190	5	0.4690	3	0.4690	3
	9000	0.9530	5	0.5630	4	0.9220	6	0.5620	4	0.5630	3
	10 000	1.0210	4	0.6580	4	1.0150	6	0.6250	4	0.6090	3
Extended Rosenbrock	1000	0.5780	33	0.7970	48	1.2820	41	0.6440	38	0.9370	43
	1200	0.5150	27	0.9310	40	1.0620	32	1.1250	59	0.9680	30
	1500	0.5620	25	0.8590	32	1.3290	78	3.7650	150	1.3440	60
	1600	0.6720	27	1.0310	31	1.3600	71	1.7500	65	1.4680	62
Raydan 1	1200	0.0940	7	0.1610	8	0.8220	20	0.1090	9	0.1090	22
	2000	0.1410	8	1.3470	25	0.7340	15	0.3900	12	0.7500	17
	2400	0.1250	6	0.8410	18	0.7970	14	0.3130	10	0.3280	11
	4000	0.5020	8	1.4220	14	1.4370	11	9.1100	58	14.7340	126
Qing	1400	0.4530	7	17.6480	397	0.9220	14	21.5030	437	57.1440	1278
	1500	0.4840	7	19.4370	401	68.733	838	99.5020	1406	74.1340	925
	1600	0.5000	7	23.2280	429	79.555	1226	110.789	1448	90.3330	1360
Rastrigin	1200	7.2030	197	4.8280	93	3.5000	93	7.6400	197	7.6560	197
	1300	0.2660	10	1.7340	46	3.4680	87	2.9530	74	3.8100	87
	1900	0.2970	6	0.6250	16	0.4690	8	3.5420	66	0.6250	16
Penalty	1950	0.4380	9	8.0710	134	0.5160	10	1.6040	37	2.6570	51
	2000	0.4530	10	0.7160	17	0.4690	10	0.9220	23	1.4530	32
	3000	0.6720	7	0.7030	8	0.8240	8	3.9380	48	12.6510	142
Perquadratic	1300	12.3280	512	14.2190	583	Inf	Inf	13.7750	537	18.4460	703
	1600	18.2630	557	21.4920	686	Inf	Inf	25.8070	746	31.7860	915
Hager	1200	10.3030	234	15.6870	354	Inf	Inf	11.1720	268	8.2500	424
	1700	28.3630	10	39.9520	606	Inf	Inf	79.4380	1676	Inf	Inf
Extended Himmelblau	2000	0.4840	6	0.8440	28	Inf	Inf	Inf	Inf	0.7030	19
	4000	0.7970	6	1.3440	19	Inf	Inf	Inf	Inf	1.3750	20
	5000	1.0150	6	1.9840	22	Inf	Inf	Inf	Inf	1.5630	19
	6000	1.1560	6	1.9210	18	Inf	Inf	Inf	Inf	1.8750	17

parameters (mean, mode, median). The problem of estimating the mode function of a probability density function (p.d.f.) has taken considerable attention in the past for both independent and dependent data, and a number of distinguished papers deal with this topic. For example, Parzen [19] and Eddy [10] for estimation of the unconditional mode in the independent and identically distributed (i.i.d.) case.

In this section, it is considered that the problem of estimating the mode of a multivariate unimodal probability density f with support in $\mathbb{R}^n$ from i.i.d. standard normal random variables $X_1, \ldots, X_n$ with common probability density function f. This problem has been investigated in numerous papers. To quote a few of them, Konakov [15] and Samanta [24]. We assume that density f has an unique mode denoted by θ and defined by

$$f(\theta) = \max_{x \in \mathbb{R}^n} f(x). \tag{5.1}$$

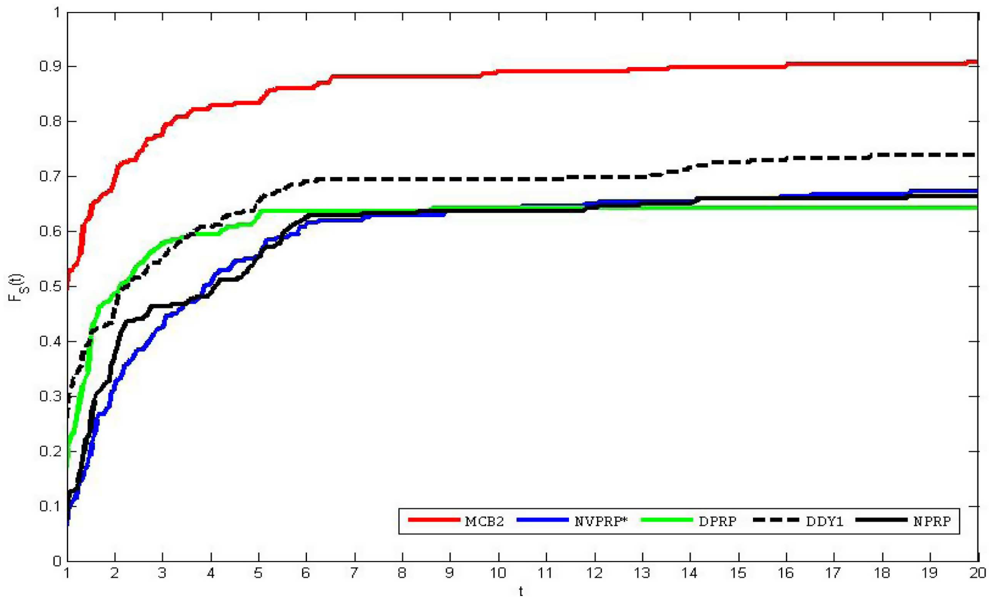

FIGURE 3. Performance profile on the CPU time (MCB2).

A kernel estimator of the mode θ is defined as the random variable $\hat{\theta}$ which maximizer the kernel estimator $f_n(x)$ of $f(x)$, that is

$$f_n\left(\hat{\theta}\right) = \max_{x\in\mathbb{R}^n} f_n(x), \tag{5.2}$$

where

$$f_n(x) = \frac{1}{nh_n^n}\sum_{i=1}^{n} K\left(\frac{x - X_i}{h_n}\right). \tag{5.3}$$

The bandwidth (h_n) is a sequence of positive real numbers which goes to zero as n goes to infinity and the kernel K is a p.d.f. on $\mathbb{R}^n$ In this simulation, we choose between two different types of kernel: while standard Gaussian kernel defined by

$$K(x) = \frac{1}{(2\pi)^{\frac{n}{2}}}\exp\left(-\frac{1}{2}\sum_{j=1}^{n} x_j^2\right),$$

and Epanechnikov kernel obtained by

$$K(x) = \left(\frac{3}{4}\right)^n \prod_{j=1}^{n}(1 - x_j^2).$$

The selection of the bandwidth h is an important and basic problem in kernel smoothing techniques. In this simulation, the optimal bandwidth by the cross-validation method is chosen.

In this context, the MCB1 and MCB2 methods are employed to solve the problem (5.2) under the SWLS technique and compare with NHS [27] and DDY1 [28] and NPRP [27] methods. According to Tables 3 and 4, it is clear that the MCB1 method is more efficient than NHS [27], DDY1 [28] methods and the MCB2 method is superior to the NPRP [27] and DDY1 [28] methods based on the number of iterations and CPU time for solving the problem (5.2).

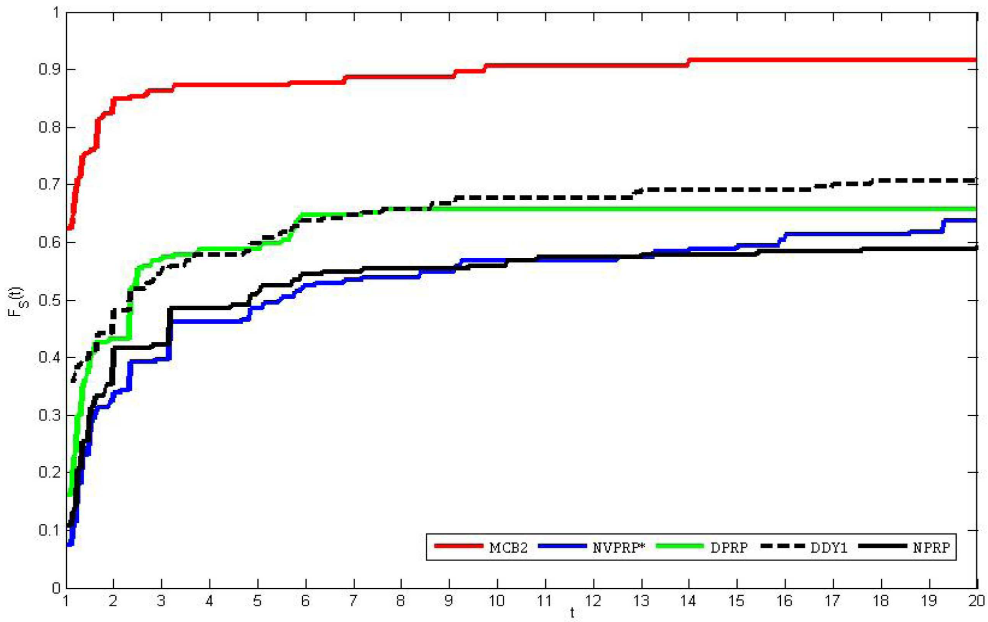

FIGURE 4. Performance profile on the number of iterations (MCB2).

TABLE 2. The simulation results of MCB2, DDY1, NVPRP*, DPRP and NPRP methods.

Method / Function	Dim	MCB2		DDY1		NVPRP*		DPRP		NPRP	
		TIME	ITR	TIME	ITR	TIME	ITR	TIME	ITR	TIME	ITR
Extended Hiebert	3000	16.5630	164	16.4840	164	6.4320	97	Inf	Inf	12.2230	197
	4000	0.7900	6	26.0870	190	7.1700	61	Inf	Inf	19.1700	201
	5000	1.0320	6	31.9470	186	Inf	Inf	Inf	Inf	24.0340	207
Penalty	1200	0.1720	7	1.9060	59	0.4060	14	0.2030	9	3.1410	91
	1400	0.3030	13	0.4370	49	1.4540	29	0.2340	8	2.6250	69
	1600	0.2180	7	35.0840	686	4.4690	101	0.2660	9	2.0000	49
Extended Himmelblau	1400	0.2920	18	0.3440	16	0.3750	19	0.2810	15	0.3750	19
	1500	0.2030	7	0.2820	12	0.3910	19	0.3130	15	0.3910	19
	1800	0.3690	16	Inf	Inf	0.4690	19	0.3750	15	0.4530	19
Quartic	805	0.6100	60	Inf	Inf	0.9690	89	0.9220	91	0.9680	89
	950	1.3990	128	Inf	Inf	2.2030	195	2.2030	194	2.6220	195
	1400	2.4360	129	Inf	Inf	3.1520	199	3.0980	199	3.8350	199
	1800	2.8910	132	Inf	Inf	4.3910	200	4.4890	200	4.4810	203
Extended Rosenbrock	1000	1.0780	55	Inf	Inf	2.6720	128	71.7950	1689	72.5990	1689
	1300	0.8440	35	Inf	Inf	7.5470	258	71.5930	1386	70.6100	1386
Diagonal 4	1800	0.0470	7	0.1090	10	0.1090	10	0.1100	10	0.1090	10
	2000	0.0930	7	Inf	Inf	0.1400	10	0.1100	10	0.1100	10
	2900	0.0410	4	0.0490	4	0.0470	4	0.0470	4	0.0460	4
Styblinski	800	0.6880	18	0.8130	21	1.9380	26	0.8120	21	1.0080	21
	950	3.5940	68	5.6720	112	5.2340	98	6.4340	114	6.3000	114
	990	3.9060	72	27.6560	468	28.600	477	30.342	495	29.655	495

Table 2. continued.

Method / Function	Dim	MCB2 TIME	MCB2 ITR	DDY1 TIME	DDY1 ITR	NVPRP* TIME	NVPRP* ITR	DPRP TIME	DPRP ITR	NPRP TIME	NPRP ITR
Double	1300	0.4250	12	0.3430	8	1.2810	40	Inf	Inf	1.3280	39
Border	6000	0.6250	5	1.0000	7	5.3240	32	Inf	Inf	7.4140	40
Arrow Up	6500	0.6560	5	1.1100	7	4.8750	31	Inf	Inf	6.1410	37
Extended White and Holst	4000	0.6310	3	1.2340	5	0.7660	4	0.7970	4	0.8130	4
Linear	600	0.0930	6	2.2500	225	2.6250	238	Inf	Inf	7.6250	995
Perturbed	2000	0.2190	6	55.4960	1007	16.9110	446	Inf	Inf	51.3910	1056
Quarticm	1200	0.2660	5	16.5750	1122	15.7800	354	Inf	Inf	15.4990	1117
Extended	100	0.0460	13	0.0630	19	Inf	Inf	Inf	Inf	8.0160	649
Tridiagonal 1	120	0.0620	13	0.0780	19	Inf	Inf	Inf	Inf	Inf	Inf
Alpine 1	400	14.1490	4	50.5650	14	103.896	25	Inf	Inf	Inf	Inf
	600	1.8630	88	1.6420	83	3.7190	134	2.3220	117	1.9420	88
Hager	800	0.1090	5	5.2350	169	Inf	Inf	3.5640	138	Inf	Inf
	1000	2.5250	95	2.6560	95	6.5310	190	3.3440	113	Inf	Inf
Chung	1200	1.2790	5	7.0460	104	6.1400	100	5.6240	30	6.3270	104
Exponential	602	0.0310	2	0.0320	3	0.0630	3	0.0470	2	0.0320	3
	1700	0.6090	16	3.0470	81	3.0320	82	3.0310	81	3.0160	81
Griewank	1800	1.4740	21	1.4380	20	1.5160	21	1.4700	20	1.6410	20
	2200	0.9380	17	5.0940	104	5.3110	105	5.0620	104	5.2420	104
Power	700	0.0930	7	50.0120	1821	Inf	Inf	0.1250	7	Inf	Inf
Schwefel1 221	2000	0.2960	9	0.2810	8	Inf	Inf	Inf	Inf	Inf	Inf
Liarwhd	110	0.0320	6	Inf	Inf	Inf	Inf	Inf	Inf	0.8900	175
Engval 1	1800	0.1400	4	0.1720	7	0.3430	7	0.3280	7	0.2810	6
Almost Perturbed	1900	0.3890	6	50.1230	408	7.0880	213	Inf	Inf	5.8750	184
Quartic	23 000	64.890	95	50.0970	39	64.6660	94	Inf	Inf	76.3750	156
Almost Perturbed	140	0.0160	3	10.7510	1398	3.7030	578	0.0320	5	4.6090	697
Quadratic	200	0.0470	3	Inf	Inf	5.0150	553	0.0470	5	5.2810	704
	1500	0.2500	7	Inf	Inf	5.0280	112	Inf	Inf	Inf	Inf
Nonscomp	1800	0.2970	7	Inf	Inf	5.0000	84	Inf	Inf	Inf	Inf
	2000	0.3280	7	Inf	Inf	5.0150	82	Inf	Inf	Inf	Inf
	950	0.2040	28	1.7670	251	Inf	Inf	Inf	Inf	Inf	Inf
Schwefel 223	1800	0.3430	29	1.3590	120	Inf	Inf	Inf	Inf	Inf	Inf
	1900	0.3600	30	3.5510	293	Inf	Inf	Inf	Inf	Inf	Inf
Alpine 2	600	2.8590	2	2.9220	2	2.8590	2	2.8590	2	2.8750	2
	1000	0.1760	8	0.2080	8	0.0690	7	Inf	Inf	0.1720	8
Raydan 2	1500	0.0780	6	148.65	1970	0.1250	8	Inf	Inf	0.2030	8
	1800	2.2720	48	2.2660	47	1.4220	31	Inf	Inf	0.2340	8
	800	0.3880	14	73.6360	1820	2.2340	84	0.3590	12	2.6250	102
Rastrigin	1100	0.2030	9	73.6980	1509	0.5940	21	0.4370	12	1.5000	46
	1200	0.1100	6	51.1870	1050	4.2650	113	0.4380	11	1.8590	52
Quadratic	1470	0.1720	7	1.8750	78	55.7450	1466	0.2190	6	0.2190	6
	470	0.0940	6	30.9690	1532	1.7650	140	3.3600	259	2.8440	222
Qing	1000	0.2190	6	3.5460	65	8.4760	253	30.2320	787	17.0160	489
	1600	2.1130	30	2.0780	28	16.2970	1231	78.9530	1231	79.3470	1193
	1200	0.1870	6	13.8430	607	22.6880	838	0.2040	7	29.3860	1075
Perquadratic	1800	2.7340	10	2.9220	115	2.1870	111	0.3590	11	2.7340	102

TABLE 2. continued.

Method / Function	Dim	MCB2		DDY1		NVPRP*		DPRP		NPRP	
		TIME	ITR	TIME	ITR	TIME	ITR	TIME	ITR	TIME	ITR
	2800	0.6090	10	7.5880	141	7.7020	142	0.7690	12	6.5620	122
Cube	1600	0.6890	7	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
	1700	0.7180	7	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
Dixon	6000	0.6570	4	0.2820	2	0.9060	5	0.2660	2	1.0620	6
Ridge	6000	0.0630	6	0.4130	66	Inf	Inf	Inf	Inf	Inf	Inf
Prod 2	1300	1.5600	6	1.5150	6	1.4220	5	0.3600	7	0.3440	7
Harkerp	400	0.1410	10	0.1400	10	17.7420	926	0.0620	9	17.7360	926
Matyas	2	0.0150	6	0.0160	7	0.0160	7	0.0160	7	0.0160	7
Leon	2	1.1720	82	0.9260	80	0.2190	47	0.2220	29	0.2030	29
Sphere	1400	0.1650	11	0.2020	12	0.1720	14	0.1720	14	0.1880	14
	1900	0.1720	11	0.3280	17	0.2190	14	0.2500	14	0.2190	14
Raydan 1	80	0.1620	90	0.0780	55	0.0780	55	0.1090	100	0.1090	88
	125	0.1870	65	0.2030	77	0.2660	77	0.2970	136	0.3130	155
	2000	0.4370	17	0.4370	17	0.3130	14	0.2970	14	0.3230	14

TABLE 3. The simulation result of MCB1, DDY1 and NHS methods for solving problem (5.2).

Kernel	Initial points	Dim	MCB1		DDY1		NHS	
			ITR	TIME	ITR	TIME	ITR	TIME
Epanechnikov	$(0.545, \ldots, 0.545)$	8	113	1.5350	144	1.8120	131	1.7350
		10	18	0.2390	125	2.3280	44	0.8440
		12	15	0.3880	14	0.3900	63	1.7280
		14	6	0.1870	8	0.2350	7	0.2030
		16	5	0.2310	6	0.2560	9	0.3840
		20	9	0.5250	7	0.4530	6	0.3910
		40	3	0.3900	4	0.6560	4	0.7820
Gaussian	$(0.545, \ldots, 0.545)$	160	6	16.466	87	228.17	7	17.686
		170	5	12.066	33	99.078	63	80.500
		190	14	45.551	20	69.828	15	47.147
		180	14	46.782	19	60.440	18	65.532
		270	4	30.020	28	204.83	48	350.62
		290	24	162.81	13	109.75	18	151.87
		320	12	127.57	15	155.18	28	289.17
		340	9	107.19	10	111.31	16	236.39
	$(0.01, \ldots, 0.01)$	26	36	2.6560	106	8.8670	975	67.547
		32	86	8.9690	44	4.3600	86	8.9830
		37	51	7.4690	112	14.891	57	7.7770
		40	11	1.9220	80	11.133	85	12.175
		48	64	13.343	206	42.625	84	17.657
		50	10	2.2030	37	8.2510	54	12.171
		58	4	1.0310	8	2.7500	43	15.047
		60	14	10.968	31	10.000	13	4.2660

TABLE 4. The simulation result of MCB2, DDY1 and NPRP methods for solving problem (5.2).

Kernel	Initial points	Dim	MCB2		DDY1		NPRP	
			ITR	TIME	ITR	TIME	ITR	TIME
Epanechnikov	(0.545, ..., 0.545)	06	77	0.5780	106	0.7970	79	0.6250
		08	142	1.5780	144	1.6100	140	1.5400
		10	35	0.4840	55	0.9060	199	3.2660
		12	26	0.6410	72	1.7030	109	2.5460
		20	4	0.1720	10	0.5310	8	0.4060
		30	3	0.4060	4	0.4210	4	0.4370
		100	4	7.1720	4	12.688	4	9.5160
Gaussian	(0.545, ..., 0.545)	100	55	34.605	159	156.67	98	97.4640
		120	48	72.553	Inf	Inf	180	263.802
		140	6	5.8730	8	12.793	35	69.2470
		150	40	36.454	87	205.79	31	70.6320
		170	15	34.534	66	205.12	63	182.250
		310	4	42.396	Inf	Inf	Inf	Inf
		340	4	36.493	13	160.27	Inf	Inf
	(0.01, ..., 0.01)	40	27	4.2340	81	12.938	97	15.2500
		46	11	2.2810	28	5.7810	31	06.4460
		48	12	2.8280	24	5.9840	78	18.2660
		52	18	4.7340	94	24.703	40	10.5470
		55	10	4.7820	30	11.203	23	08.5630
		57	9	2.8440	35	11.141	55	17.3440
		62	13	4.8250	13	4.8440	7	0 2.6250
		68	9	3.6090	12	4.7810	8	0 3.2030

6. Conclusion

This paper has presented two modified conjugate gradient methods, that is, MCB1 and MCB2 methods. Under the SWLS condition the sufficient descent condition of the MCB1 method has been established. An attractive property of the MCB2 method is that it generates a sufficient descent condition, regardless of the line search. The global convergence properties of the MCB1 and MCB2 methods have been established under the SWLS conditions.

From the statistical results obtained by the first comparison technique in Figures 1 and 3, it is clear that the average CPU time of the MCB1 and MCB2 methods are approximately equal.

From Figures 2 and 4, the MCB1 method is slightly more effective than the MCB2 method, with respect to the number of iterations.

The final conclusion is that the proposed methods are more efficient than some existing methods. The practical applicability of MCB1 and MCB2 methods is also explored in nonparametric estimation of the mode function.

Acknowledgements. The authors are very grateful to the anonymous referees for their valuable comments and useful suggestions, which improved the quality of this paper.

References

[1] M. Al-Baali, Descent property and global convergence of the Fletcher–Reeves method with inexact line search. *IMA J. Numer. Anal.* **5** (1985) 121–124.

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

[3] A.M. Awwal, I.M. Sulaiman, M. Malik, M. Mamat, P. Kumam and K. Sitthithakerngkiet, A spectral RMIL conjugate gradient method for unconstrained optimization with applications in portfolio selection and motion control. *IEEE Access* **9** (2021) 75398–75414.

[4] I. Bongartz, A.R. Conn, N. Gould and P.L. Toint, Constrained and unconstrained testing environment. *ACM Trans. Math. Softw.* **21** (1995) 123–160.

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

[6] 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.

[7] Y.H. Dai, J.Y. Han, G.H. Liu, D.F. Sun, X. Yin and Y. Yuan, Convergence properties of nonlinear conjugate gradient methods. *SIAM J. Optim.* **10** (1999) 348–358.

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

[9] X.W. Du, P. Zhang and W. Ma, Some modified conjugate gradient methods for unconstrained optimization. *J. Comput. Appl. Math.* **305** (2016) 92–114.

[10] W.F. Eddy, Optimum kernel estimators of the mode. *Ann. Stat.* **8** (1980) 870–882.

[11] R. Fletcher, Practical Methods of Optimization, 2nd edition. Wiley, New York (1987).

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

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

[14] H. Huang, Z. Wei and S. Yao, The proof of the sufficient descent condition of the Wei–Yao–Liu conjugate gradient method under the strong Wolfe–Powell line search. *Appl. Math. Comput.* **189** (2007) 1241–1245.

[15] V.D. Konakov, On the asymptotic normality of the mode of multidimensional distributions. *Theory Probab. Appl.* **19** (1974) 794–799.

[16] J.K. Liu and S.J. Li, New hybrid conjugate gradient method for unconstrained optimization. *Appl. Math. Comput.* **245** (2014) 36–43.

[17] Y. Liu and C. Storey, Efficient generalized conjugate gradient algorithms. *Theory JOTA* **69** (1991) 129–137.

[18] G. Ma, H. Lin, W. Jin and D. Han, Two modified conjugate gradient methods for unconstrained optimization with applications in image restoration problems. *J. Appl. Math. Comput.* **68** (2022) 4733–4758.

[19] E. Parzen, On estimating probability density function and mode. *Ann. Math. Stat.* **33** (1962) 1065–1076.

[20] E. Polak and G. Ribière, Note sur la convergence de directions conjuguée. *Rev. Francaise Informat Recherche Operationelle* **16** (1969) 35–43.

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

[22] M.J.D. Powell, Nonconvex minimization calculations and the conjugate gradient method. *Lect. Notes Math.* **1066** (1984) 122–141.

[23] T.W. Sager, An iterative method for estimating a multivariate mode and isopleth. *J. Amer. Statist. Assoc.* **74** (1975) 329–339.

[24] M. Samanta, Nonparametric estimation of the mode of a multivariate density. *South Afr. Stat. J.* **7** (1973) 109–117.

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

[26] S. Yao, Z. Wei and H. Huang, A notes about WYL's conjugate gradient method and its applications. *Appl. Math. Comput.* **191** (2007) 381–388.

[27] L. Zhang, An improved Wei–Yao–Liu nonlinear conjugate gradient method for optimization computation. *Appl. Math. Comput.* **6** (2009) 2269–2274.

[28] Z. Zhu, D. Zhang and S. Wang, Two modified DY conjugate gradient methods for unconstrained optimization problems. *Appl. Math. Comput.* **373** (2020) 125004.

[29] G. Zoutendijk, Nonlinear programming computational methods. *J. Integer Nonlinear Program.* (1970) 37–86.

[30] N. Zullpakkal, N. Aini, N.H.A. Ghani, N.S. Mohamed, N. Idalisa and M. Rivaie, Covid-19 data modelling using hybrid conjugate gradient method. *J. Inf. Optim. Sci.* **43** (2022) 837–853.