A New Modified Nonlinear Conjugate Gradient Method with Sufficient Descent Property for Unconstrained Optimization

Minglei Fang, Min Wang*, Defeng Ding, Yuting Sheng

***Abstract*—Motivated by some well-known conjugate gradient methods, in this paper, we propose a new hybrid conjugate gradient method for unconstrained optimization. Without any dependence on line search, the new method generates sufficient descent property. Under some mild conditions, we prove that the new method with MWWP line search is globally convergent. Our numerical experiments show the effectiveness and robustness of the new modified method in comparison with three exiting algorithms.**

***Index Terms*—unconstrained optimization, wolfe line search, sufficient descent, global convergence.**

I. Introduction

CONJUGATE gradient method due to its simple iterations and lower storage requirements, is considered effective iterative method for solving large-scale optimization.

$$\min f(x), \quad x \in R^n, \tag{1}$$

where objective function $f(x) : R^n \to R$ is continuously differentiable and its gradient $h(x) \triangleq \nabla f(x)$ is available.

To solve problem (1), given an initial guess $x_1 \in R^n$, the conjugate gradient method usually generates a sequence $\{x_k\}$ defined by

$$x_{k+1} = x_k + \alpha_k t_k, \quad k = 1, 2, \cdots, \tag{2}$$

where $\alpha_k > 0$ is a steplength computed by carrying out some suitable exact or inexact line search, and t_k is the search direction generated by:

$$t_k = \begin{cases} -h_k & k = 1, \\ -h_k + \beta_k t_{k-1} & k \geq 2, \end{cases} \tag{3}$$

where $h_k = \nabla f(x_k)$ and β_k is an important scalar parameter.

Manuscript received November 03, 2022; revised June 5, 2023. This work has been partially supported by the Key Program of University Natural Science Research Fund of Anhui Province (no. KJ2021A0451) and Anhui Provincial Natural Science Foundation (no.2008085MA01).

M. L. Fang is an assistant professor of the School of Mathematics and Big Data, Anhui University of Science and Technology, Huainan, 232001, PR China(e-mail: fmlmath@aust.edu.cn).

*corresponding author. M. Wang is a postgraduate student of the School of Mathematics and Big Data, Anhui University of Science and Technology, Huainan, 232001, PR China.(e-mail: 2020201146@edu.aust.cn).

D. F. Ding is a postgraduate student of the School of Mathematics and Big Data, Anhui University of Science and Technology, Huainan, 232001, PR China.(e-mail: 2021201353@edu.aust.cn).

Y. T. Sheng is a postgraduate student of the School of Mathematics and Big Data, Anhui University of Science and Technology, Huainan, 232001, PR China.(e-mail: 805015354@qq.com).

In the inexact line search, the step length α_k is very important for global convergence of conjugate gradient methods, the step length α_k is usually computed by the Weak Wolfe line search

$$f(x_k + \alpha_k t_k) \leq f(x_k) + \delta \alpha_k h_k^T t_k, \tag{4}$$

$$h(x_k + \alpha_k t_k)^T t_k \geq \sigma h_k^T t_k, \tag{5}$$

or the Strong Wolfe line search

$$f(x_k + \alpha_k t_k) \leq f(x_k) + \delta \alpha_k h_k^T t_k, \tag{6}$$

$$|h(x_k + \alpha_k t_k)^T t_k| \leq -\sigma h_k^T t_k, \tag{7}$$

where parameters δ and σ satisfy $0 < \delta < \sigma < 1$.

Generally, with a distinct choice of the parameter β_k , give rise to distinct CG methods with quite different computational efficiency and convergence properties, the obtained method has different numerical performance. The parameter β_k can be divided into the following two categories.

The first category includes Fletcher and Reeves (FR)[1], Dai and Yuan (DY)[2] and Conjugate Descent (CD)[3]:

$$\beta_k^{FR} = \frac{||h_k||^2}{||h_{k-1}||^2}, \beta_k^{DY} = \frac{||h_k||^2}{t_{k-1}^T y_{k-1}}, \beta_k^{CD} = \frac{||h_k||^2}{-h_{k-1}^T t_{k-1}}.$$

The other category includes Hestenes and Stiefel (HS)[4] , Polak-Ribière-Polyak (PRP)[5], [6] and Liu and Storey (LS)[7]

$$\beta_k^{HS} = \frac{h_k^T y_{k-1}}{t_{k-1}^T y_{k-1}}, \beta_k^{PRP} = \frac{h_k^T y_{k-1}}{||h_{k-1}||^2}, \beta_k^{LS} = \frac{h_k^T y_{k-1}}{-h_{k-1}^T t_{k-1}},$$

where $|| \cdot ||$ stands for the Euclidean norm and $y_{k-1} = h_k - h_{k-1}$.

Generally speaking, the conjugate gradient methods of the first category have the common numerator $||h_k||^2$ and global convergence properties. However, they may have poor numerical performance due to jamming. In contrast to those of the first category, the conjugate gradient methods of the other category may fail to converge, but they automatically adjust the parameter β_k to avoid the jamming problem. So, they have better computational performance.

In recent years, based on the above six formulas and their complication, many scholars have devoted themselves to finding an improved conjugate gradient method that has both good convergence and good numerical effects[8], [9], [10], [11].

In[12], Wei et al. gave a modified HS formula for computing β_k as follows:

$$\beta_k^{HS_*} = \frac{h_k^T(h_k - \frac{h_k^T h_{k-1}}{||h_{k-1}||^2} h_{k-1})}{t_{k-1}^T(h_k - h_{k-1})}.$$

Under the strong Wolfe line search with the parameter $\sigma < \frac{1}{2}$, the authors proved that this method satisfies the sufficient descent condition and converges globally.

In[13], Jiang et al. compared the difference in the denominator between the MS and PRP formulas and replaces the denominator in the PRP, and the constructed a value has a better numerical effect.

By modifying the DY method, Huang[14] proposed the MDY, in which

$$\beta_k^{MDY} = \frac{h_k^T(h_k - \frac{h_k^T t_{k-1}}{||t_{k-1}||^2} t_{k-1})}{t_{k-1}^T(h_k - h_{k-1})}.$$

If using exact line search, then $\beta_k^{MDY} = \beta_k^{DY}$.

Based on the ideas on the hybrid methods[15], [16], A novel hybrid conjugate gradient method was put out by Jian et al.[17].

$$\beta_k^N = \frac{||h_k||^2 - \max\{0, \frac{||h_k||}{||h_{k-1}||} h_k^T h_{k-1}\}}{\max\{||h_{k-1}||^2, t_{k-1}^T(h_k - h_{k-1})\}}.$$

Based on works[17], [18], [19], Jiang et al.[20] updated the HS conjugate gradient method's parameter formula, in which

$$\beta_k^{new} = \frac{||h_k||^2 - \frac{h_k^T t_{k-1}}{||t_{k-1}||||h_{k-1}||} h_k^T h_{k-1}}{\mu \max\{t_{k-1}^T(h_k - h_{k-1}), |h_k^T t_{k-1}|\}} (\mu > 2).$$

Furthermore, many scholars have proposed efficient hybrid conjugate gradient methods which have nice convergence properties and excellent numerical performance in the past decades [21], [22], [23], [24], [25], [26].

In this paper, we consider an improved parameter β_k^{MH} as follows:

$$\beta_k^{MH} = \frac{||h_k||^2 - \mu_1 \cdot \max\{\frac{h_k^T t_{k-1}}{||h_{k-1}||||t_{k-1}||} h_k^T h_{k-1}, \frac{h_k^T t_{k-1}}{||t_{k-1}||^2} h_k^T t_{k-1}\}}{\max\{t_{k-1}^T(h_k - \mu_2 h_{k-1}), ||h_{k-1}||^2 + \mu_2 |h_k^T t_{k-1}|\}}, \tag{8}$$

where $0 < \mu_1 < 1, \mu_2 > 1$.

Based on formula (8), combining with the MWWP line search, we prove the global convergence of our algorithm. The numerical results also show the feasibility and effectiveness of the proposed method.

This paper is structured as follows. In Sect. 2, we describe our algorithm in detail and discuss its sufficient descent property. In Sect. 3, under MWWP line search, the global convergence property of the proposed method is established. A large amounts of numerical results comparing with other conjugate gradient Methods are reported in Sect. 4. Finally, we have a conclusion section.

II. Algorithm and its property

Based on (8), we introduce our hybrid conjugate gradient algorithm as follows:

Algorithm MH

Initialization. Given any initial point $x_1 \in R^n$ and parameters $\varepsilon \geq 0, \delta \in (0, \frac{1}{2}), \delta_1 \in (0, \delta)\sigma \in (\delta, 1)$, $0 < \mu_1 < 1$, $\mu_2 > 1$. Let $t_1 = -h_1, k := 1$.

Step 1. If $||h_k|| \leq \varepsilon$, then stop. If not, proceed to step 2.

Step 2. Compute α_k by the modified Weak Wolfe line search(MWWP)[27].

$$f(x_k + \alpha_k t_k) \leq f(x_k) + \delta \alpha_k h_k^T t_k + \alpha_k \min(-\delta_1 h_k^T t_k, \delta \tfrac{\alpha_k}{2} \|t_k\|^2), \tag{9}$$

$$h(x_k + \alpha_k t_k)^T t_k \geq \sigma h_k^T t_k + \min(-\delta_1 h_k^T t_k, \delta \alpha_k \|t_k\|^2), \tag{10}$$

Step 3. Let $x_{k+1} = x_k + \alpha_k t_k$, compute $h_{k+1} = h(x_{k+1})$ and β_{k+1}^{MH} by (8).

Step 4. Let $t_{k+1} = -h_k + \beta_k^{MH} t_k$. Set $k := k + 1$, go to Step 1.

The following lemma shows that the search direction generated by the algorithm framework MH satisfies the sufficient descent condition under the MWWP line search.

Lemma 2.1: If the search direction t_k is generated by the Algorithm MH, then the search direction t_k satisfies the sufficient descent condition:

$$h_k^T t_k \leq -c||h_k||^2 \ , \ \forall k \geq 1, \tag{11}$$

where $0 < c = 1 - \frac{1}{\mu_2} < 1$.

Proof. Using mathematical induction, for $k = 1$, it is easy to know that

$$h_1^T t_1 = -||h_1||^2 \leq -c||h_1||^2.$$

The conclusion of the Lemma 2.1 holds. Suppose that $h_{k-1}^T t_{k-1} \leq -c||h_{k-1}||^2$ also holds for $k - 1$ and $k > 2$. Now, we prove that $h_k^T t_k \leq -c||h_k||^2$ holds for k.

Let φ_k be the angle between h_k and t_{k-1} and ξ_k be the angle between h_k and h_{k-1}, respectively.

It follows that

$$\cos\varphi_k \cos\xi_k = \frac{h_k^T t_{k-1}}{||h_k||||t_{k-1}||} \cdot \frac{h_k^T h_{k-1}}{||h_k||||h_{k-1}||},$$

$$\cos^2\varphi_k = \frac{(h_k^T t_{k-1})^2}{||h_k||^2||t_{k-1}||^2}.$$

Combining (8), we have

$$\beta_k^{MH} = \frac{||h_k||^2\{1 - \mu_1 \cdot \max\{\cos\varphi_k \cos\xi_k, \cos^2\varphi_k\}\}}{\max\{t_{k-1}^T(h_k - \mu_2 h_{k-1}), ||h_{k-1}||^2 + \mu_2 |h_k^T t_{k-1}|\}}.$$

Since $-1 \leq \cos\varphi_k \cos\xi_k \leq 1$ and $0 \leq \cos^2\varphi_k \leq 1$, if $\cos\varphi_k \cos\xi_k > \cos^2\varphi_k$, we have $0 \leq \max\{\cos\varphi_k \cos\xi_k, \cos^2\varphi_k\} = \cos\varphi_k \cos\xi_k \leq 1$. For simplicity of presentation, let

$$t = \max\{\cos\varphi_k \cos\xi_k, \cos^2\varphi_k\}.$$

It is clear that

$$\beta_k^{MH}=\frac{||h_k||^2(1-\mu_1\cdot t)}{\max\{t_{k-1}^T(h_k-\mu_2 h_{k-1}),||h_{k-1}||^2+\mu_2|h_k^T t_{k-1}|\}}, \tag{12}$$

where $0 \leq t \leq 1$.

If $h_k^T t_{k-1} = 0$, and combining the values of μ_1 and μ_2, we can derive the following result:

$$\begin{aligned} h_k^T t_k &= -||h_k||^2 + \beta_k t_{k-1}^T h_k \\ &= -||h_k||^2 + \\ &\frac{||h_k||^2(1-\mu_1\cdot t)\cdot h_k^T t_{k-1}}{\max\{t_{k-1}^T(h_k-\mu_2 h_{k-1}),||h_{k-1}||^2+\mu_2|h_k^T t_{k-1}|\}} \\ &= -||h_k||^2 \leq -(1-\frac{1}{\mu_2})||h_k||^2. \end{aligned}$$

If $h_k^T t_{k-1} \neq 0$, we divide the proof into two following cases.

(i) When $t_{k-1}^T(h_k-\mu_2 h_{k-1}) \geq ||h_{k-1}||^2+\mu_2|h_k^T t_{k-1}|$, we get

$$\begin{aligned} h_k^T t_k &= -||h_k||^2 + \beta_k h_k^T t_{k-1} \\ &= -||h_k||^2 + \frac{||h_k||^2(1-\mu_1\cdot t)}{t_{k-1}^T(h_k-\mu_2 h_{k-1})} h_k^T t_{k-1} \\ &\leq -||h_k||^2 + \frac{||h_k||^2(1-\mu_1\cdot t)}{t_{k-1}^T(h_k-\mu_2 h_{k-1})} |h_k^T t_{k-1}| \\ &\leq -||h_k||^2 + \frac{||h_k||^2(1-\mu_1\cdot t)}{||h_{k-1}||^2+\mu_2|h_k^T t_{k-1}|} |h_k^T t_{k-1}| \\ &\leq -||h_k||^2 + \frac{||h_k||^2(1-\mu_1\cdot t)}{\mu_2|h_k^T t_{k-1}|} |h_k^T t_{k-1}| \\ &\leq -(1-\frac{1}{\mu_2})||h_k||^2. \end{aligned}$$

(ii) When $t_{k-1}^T(h_k-\mu_2 h_{k-1}) < ||h_{k-1}||^2+\mu_2|h_k^T t_{k-1}|$, we have that

$$\begin{aligned} h_k^T t_k &= -||h_k||^2 + \beta_k h_k^T t_{k-1} \\ &= -||h_k||^2 + \frac{||h_k||^2(1-\mu_1\cdot t)}{||h_{k-1}||^2+\mu_2|h_k^T t_{k-1}|} h_k^T t_{k-1} \\ &\leq -||h_k||^2 + \frac{||h_k||^2(1-\mu_1\cdot t)}{||h_{k-1}||^2+\mu_2|h_k^T t_{k-1}|} |h_k^T t_{k-1}| \\ &\leq -||h_k||^2 + \frac{||h_k||^2(1-\mu_1\cdot t)}{\mu_2|h_k^T t_{k-1}|} |h_k^T t_{k-1}| \\ &\leq -(1-\frac{1}{\mu_2})||h_k||^2. \end{aligned}$$

Thus, the lemma 2.1 holds.

Lemma 2.2: If the step size α_k satisfies the MWWP line search conditions (9) and (10), for all $k \geq 1$, we have

$$0 \leq \beta_k^{MH} \leq \frac{h_k^T t_k}{h_{k-1}^T t_{k-1}}. \tag{13}$$

Proof. First, let us prove left side of the inequality, we have from (12) that the numerator of parameter β_k^{MH} is greater than or equal to 0. Further, we just need to prove that the denominator is greater than 0 by considering the following two cases.

(i) if $t_{k-1}^T(h_k-\mu_2 h_{k-1}) \geq ||h_{k-1}||^2+\mu_2|h_k^T t_{k-1}|$, we have from (10) that

$$\begin{aligned} &\max\{t_{k-1}^T(h_k-\mu_2 h_{k-1}),||h_{k-1}||^2+\mu_2|h_k^T t_{k-1}|\} \\ &= t_{k-1}^T(h_k-\mu_2 h_{k-1}) \\ &\geq \sigma h_{k-1}^T t_{k-1} + \min(-\delta_1 h_k^T t_k, \delta\alpha_k\|t_k\|^2) - \mu_2 h_{k-1}^T t_{k-1} \\ &> (\sigma-\mu_2) h_{k-1}^T t_{k-1} > 0. \end{aligned}$$

(ii) if $t_{k-1}^T(h_k-\mu_2 h_{k-1}) < ||h_{k-1}||^2+\mu_2|h_k^T t_{k-1}|$, we have

$$\begin{aligned} &\max\{t_{k-1}^T(h_k-\mu_2 h_{k-1}),||h_{k-1}||^2+\mu_2|h_k^T t_{k-1}|\} \\ &= ||h_{k-1}||^2+\mu_2|h_k^T t_{k-1}| \geq t_{k-1}^T(h_k-\mu_2 h_{k-1}) > 0. \end{aligned}$$

Next, we prove the right side of the inequality by considering the following two cases.

(i) if $t_{k-1}^T(h_k-\mu_2 h_{k-1}) \geq ||h_{k-1}||^2+\mu_2|h_k^T t_{k-1}|$, then

$$\beta_k^{MH} = \frac{||h_k||^2(1-\mu_1\cdot t)}{t_{k-1}^T(h_k-\mu_2 h_{k-1})}.$$

Furthermore, we know that

$$\begin{aligned} h_k^T t_k &= -||h_k||^2 + \beta_k^{MH} h_k^T t_{k-1} \\ &= -||h_k||^2 + \frac{||h_k||^2(1-\mu_1\cdot t)}{t_{k-1}^T(h_k-\mu_2 h_{k-1})} h_k^T t_{k-1} \\ &= -||h_k||^2 + \frac{||h_k||^2(1-\mu_1\cdot t) h_k^T t_{k-1}}{t_{k-1}^T(h_k-\mu_2 h_{k-1})} \\ &+ \frac{-\mu_2||h_k||^2(1-\mu_1\cdot t) h_{k-1}^T t_{k-1}}{t_{k-1}^T(h_k-\mu_2 h_{k-1})} \\ &+ \frac{\mu_2||h_k||^2(1-\mu_1\cdot t) h_{k-1}^T t_{k-1}}{t_{k-1}^T(h_k-\mu_2 h_{k-1})} \\ &= -||h_k||^2 + ||h_k||^2(1-\mu_1\cdot t) \\ &+ \frac{\mu_2||h_k||^2(1-\mu_1\cdot t) h_{k-1}^T t_{k-1}}{t_{k-1}^T(h_k-\mu_2 h_{k-1})} \\ &= -\mu_1\cdot t||h_k||^2 + \frac{\mu_2||h_k||^2(1-\mu_1\cdot t)}{t_{k-1}^T(h_k-\mu_2 h_{k-1})} h_{k-1}^T t_{k-1} \\ &\leq \frac{\mu_2||h_k||^2(1-\mu_1\cdot t)}{t_{k-1}^T(h_k-\mu_2 h_{k-1})} h_{k-1}^T t_{k-1} \\ &= \mu_2 \beta_k^{MH} h_{k-1}^T t_{k-1}. \end{aligned} \tag{14}$$

(ii) if $t_{k-1}^T(h_k-\mu_2 h_{k-1}) < ||h_{k-1}||^2+\mu_2|h_k^T t_{k-1}|$, then

$$\beta_k^{MH} = \frac{||h_k||^2(1-\mu_1\cdot t)}{||h_{k-1}||^2+\mu_2|h_k^T t_{k-1}|}.$$

In the similar vein, we get

$$
\begin{aligned}
&h_k^T t_k = -||h_k||^2 + \beta_k^{MH} h_k^T t_{k-1} \\
&= -||h_k||^2 + \frac{||h_k||^2(1-\mu_1 \cdot t)}{||h_{k-1}||^2 + \mu_2 |h_k^T t_{k-1}|} h_k^T t_{k-1} \\
&\leq -||h_k||^2 + \frac{||h_k||^2(1-\mu_1 \cdot t)}{||h_{k-1}||^2 + \mu_2 |h_k^T t_{k-1}|} \\
&\cdot \frac{(||h_{k-1}||^2 + \mu_2 |h_k^T t_{k-1}| + \mu_2 h_{k-1}^T t_{k-1})}{||h_{k-1}||^2 + \mu_2 |h_k^T t_{k-1}|} \\
&= -||h_k||^2 + ||h_k||^2(1-\mu_1 \cdot t) \\
&+ \frac{||h_k||^2(1-\mu_1 \cdot t)}{||h_{k-1}||^2 + \mu_2 |h_k^T t_{k-1}|} \mu_2 h_{k-1}^T t_{k-1} \\
&= -\mu_1 \cdot t ||h_k||^2 + \frac{||h_k||^2(1-\mu_1 \cdot t)}{||h_{k-1}||^2 + \mu_2 |h_k^T t_{k-1}|} \mu_2 h_{k-1}^T t_{k-1} \\
&\leq \frac{\mu_2 ||h_k||^2(1-\mu_1 \cdot t)}{||h_{k-1}||^2 + \mu_2 |h_k^T t_{k-1}|} h_{k-1}^T t_{k-1} \\
&= \mu_2 \beta_k^{MH} h_{k-1}^T t_{k-1}.
\end{aligned} \tag{15}
$$

Combining the Lemma 2.2, dividing both sides of (14) and (15) by $h_{k-1}^T t_{k-1}$, it is easy to get

$$\mu_2 \beta_k^{MH} \leq \frac{h_k^T t_k}{h_{k-1}^T t_{k-1}}.$$

Because of $\mu_2 > 1$, which shows $\beta_k^{MH} \leq \frac{h_k^T t_k}{h_{k-1}^T t_{k-1}}$, $\forall k \geq 1$. The proof is completed.

III. Global Convergence of Algorithm

In this section, we are going to analyse the global convergence of the proposed algorithm. The following assumptions are needed.

H 3.1: The level set $\Lambda = \{x \in R^n | f(x) \leq f(x_1)\}$ is bounded below for all $x_1 \in R^n$.

H 3.2: The objective function $f(x)$ is continuously differentiable and its gradient $h(x)$ is Lipschitz continuous on an open set U containing the level set Λ, i.e., there exists a constant $L > 0$ such that

$$||h(x) - h(y)|| \leq L||x - y||, \forall x, y \in U. \tag{16}$$

A very useful result for establishing the convergence of the conjugate gradient algorithms is the famous Zoutendijk conditions:

Lemma 3.1: Suppose that Assumptions (H 3.1) and (H 3.2) hold, where the search direction t_k satisfies the descent condition $h_k^T t_k < 0$ and the stepsize α_k satisfies the MWWP line search conditions (9) and (10). Then

$$\sum_{k=1}^{\infty} \frac{(h_k^T t_k)^2}{||t_k||^2} < \infty. \tag{17}$$

Proof. From (10), it follows that,

$$
\begin{aligned}
&\sigma h_k^T t_k - h_k^T t_k \\
&\leq h_{k+1}^T t_k - \min(-\delta_1 h_k^T t_k, \delta \alpha_k \|t_k\|^2) - h_k^T t_k \\
&\leq h_{k+1}^T t_k - h_k^T t_k.
\end{aligned}
$$

Combining with (H 3.2), we derive

$$
\begin{aligned}
(\sigma - 1) h_k^T t_k &\leq \|(h_{k+1} - h_k)^T t_k\| \\
&\leq \|h_{k+1} - h_k\| \|t_k \\
&\leq L \|x_{k+1} - x_k\| \|t_k\| \\
&= \alpha_k L \|t_k\|^2.
\end{aligned} \tag{18}
$$

By $(\sigma - 1) < 0$ and (18), we have

$$\alpha_k \geq \frac{\sigma - 1}{L} \frac{h_k^T t_k}{\|t_k\|^2} > 0.$$

Therefore, from (9) and (11), we get

$$
\begin{aligned}
&f_{k+1} - f_k \leq \delta \alpha_k h_k^T t_k + \alpha_k \min(-\delta_1 h_k^T t_k, \delta \frac{\alpha_k}{2} \|t_k\|^2) \\
&\leq (\delta - \delta_1) \alpha_k h_k^T t_k \leq (\delta - \delta_1) \frac{(\sigma - 1)}{L} \frac{h_k^T t_k}{\|t_k\|^2} h_k^T t_k \\
&= (\delta - \delta_1) \frac{(\sigma - 1)}{L} \frac{(h_k^T t_k)^2}{\|t_k\|^2}.
\end{aligned} \tag{19}
$$

Summing the above inequalities for k from 1 to ∞, we have

$$
\begin{aligned}
&(\delta - \delta_1) \frac{(1 - \sigma)}{L} \sum_{k=1}^{\infty} \frac{(h_k^T t_k)^2}{\|t_k\|^2} \leq \sum_{k=1}^{\infty} (f_k - f_{k+1}) \\
&\leq (f_1 - f_2) + (f_2 - f_3) + \cdots = f_1 - f^{\infty}.
\end{aligned}
$$

By Assumptions (H 3.1) and (H 3.2), since $f(x)$ is bounded below on Λ, we obtain

$$\sum_{k=1}^{\infty} \frac{(h_k^T t_k)^2}{||t_k||^2} < \infty.$$

Theorem 3.1: Suppose that Assumptions (H 3.1) and (H 3.2) hold. Let the sequence $\{x_k\}$, $\{h_k\}$ and $\{t_k\}$ be generated by Algorithm MH. Then

$$\lim_{k \to \infty} inf ||h_k|| = 0. \tag{20}$$

Proof. By contradiction, assume (20) is not true. Due to $\lim_{k \to \infty}$ inf $\|h_k\| \neq 0$, it is evident that $\|h_k\| > 0$. Then, there exists a constant $\gamma > 0$ such that $||h_k||^2 \geq \gamma$ for all $k \geq 1$.

From (3), we have

$$t_k + h_k = \beta_k^{MH} t_{k-1}.$$

Squaring both sides above equality, we get

$$||t_k||^2 + 2h_k^T t_k + ||h_k||^2 = (\beta_k^{MH})^2 ||t_{k-1}||^2.$$

It follows that

$$||t_k||^2 = (\beta_k^{MH})^2 ||t_{k-1}||^2 - 2h_k^T t_k - ||h_k||^2.$$

Now, dividing both sides above formula by $(h_k^T t_k)^2$ and using (13), we get

$$\frac{||t_k||^2}{(h_k^Tt_k)^2} \le \frac{(h_k^Tt_k)^2}{(h_{k-1}^Tt_{k-1})^2}\frac{||t_{k-1}||^2}{(h_k^Tt_k)^2} - \frac{2h_k^Tt_k}{(h_k^Tt_k)^2} - \frac{||h_k||^2}{(h_k^Tt_k)^2}$$
$$\le \frac{||t_{k-1}||^2}{(h_{k-1}^Tt_{k-1})^2} - (\frac{||h_k||}{h_k^Tt_k} + \frac{1}{||h_k||})^2 + \frac{1}{||h_k||^2}$$
$$\le \frac{||t_{k-1}||^2}{(h_{k-1}^Tt_{k-1})^2} + \frac{1}{||h_k||^2}. \tag{21}$$

Combining with $\frac{||t_1||^2}{(h_1^Tt_1)^2} = \frac{1}{||h_1||^2}$, by a recurrence of relation (21) and $||h_k||^2 \ge \gamma$, we have

$$\frac{||t_k||^2}{(h_k^Tt_k)^2} \le \frac{||t_{k-1}||^2}{(h_{k-1}^Tt_{k-1})^2} + \frac{1}{||h_k||^2}$$
$$\le \frac{||t_{k-2}||^2}{(h_{k-2}^Tt_{k-2})^2} + \frac{1}{||h_{k-1}||^2} + \frac{1}{||h_k||^2}$$
$$\le \cdots \le \sum_{i=1}^{k}\frac{1}{||h_i||^2} \le \frac{k}{\gamma}.$$

It is easy to see that

$$\sum_{k=1}^{\infty}\frac{(h_k^Tt_k)^2}{||t_k||^2} \ge \sum_{k=1}^{\infty}\frac{\gamma}{k} = \infty.$$

Which contradicts Lemma 3.1. Therefore, the relation (20) is true. The proof is completed.

IV. Numerical Experiments

In order to evaluate the effectiveness of the modified method, we performed a comparison with three other conjugate gradient methods. Unconstrained test problems from Morè et al.[28] are solved by the MH method, the MDY method [12], the MHS method [14] and the MN method [20]. The step length $alpha_k$ for the MDY, MHS, and MN methods is obtained from the Weak Wolfe line search (4) and (5). All of the code was written in MATLAB R2015a.

The computation will be terminated for all methods if any of the following conditions are met: (i) $||h_k|| \le \varepsilon = 10^{-6}$, (ii) the number of iterations Itr>10000. The parameters are chosen as $\delta = 0.3$, $\delta_1 = 0.1$, $\sigma = 0.6$, $\mu_1 = 0.1$, $\mu_2 = 1.1$.

The comparison of data is displayed in Table 1, where "p" is the name of the test problem, "n" is the dimension of the test problem, "NF" denotes the number of function evaluations, "NG" means the number of gradient evaluations, "NI" represents the number of iterations and "Tcpu" represents the running time of the CPU in seconds for calculating the test questions. Specially, "-" means that the test problem cannot be successfully solved within 10000 iterations.

In addition, we analyse the effectiveness of these four techniques using the performance profiles developed by Dolan and Morè [29]. Let S be a set of n_s comparison solvers and P be a set of n_p test problems. For each problem $p \in P$ and solver $s \in S$, denote $t_{p,s}$ is the number of function gradients, iterations or CPU time required

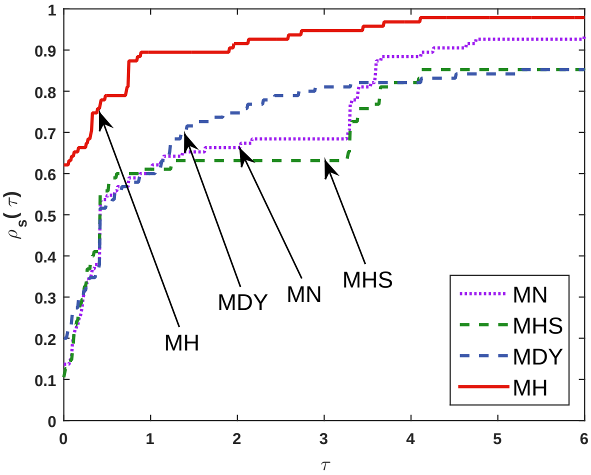

Fig. 1: Performance profile on the number of function evaluation

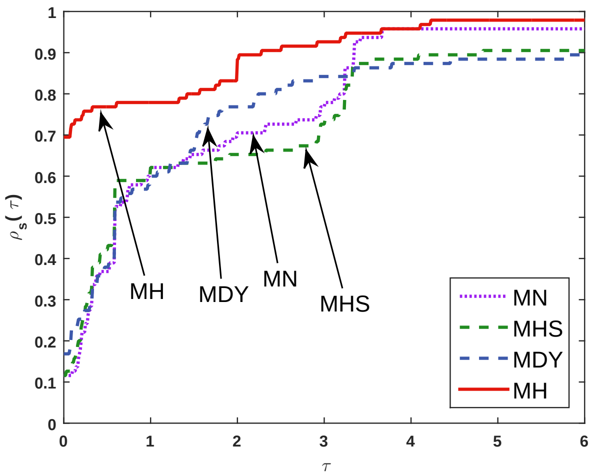

Fig. 2: Performance profile on the number of gradient evaluation

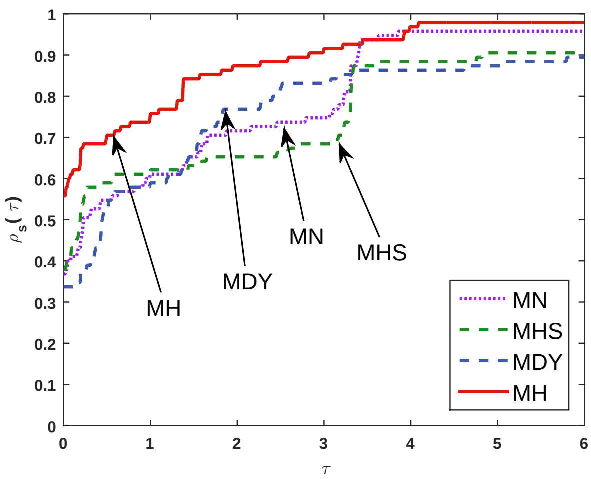

Fig. 3: Performance profile on the number of iteration

TABLE I: Test results of the MH , MDY , MHS and MN methods

p	n	MH				MDY				MHS				MN			
		NF	NG	NI	Tcpu	NF	NG	NI	Tcpu	NF	NG	NI	Tcpu	NF	NG	NI	Tcpu
ROSE	2	151	56	41	0.027	275	109	67	0.029	152	57	29	0.027	173	93	56	0.027
FROTH	2	174	73	53	0.035	161	61	39	0.017	45	21	11	0.004	185	76	49	0.021
BADSCP	2	55	4	3	0.013	101	5	2	0.008	101	5	2	0.009	101	5	2	0.009
BADSCB	2	139	19	13	0.012	-	-	-	-	-	-	-	-	104	23	13	0.013
BEALE	2	59	28	21	0.020	191	110	72	0.024	63	34	22	0.008	44	24	15	0.009
JENSAM	2	52	3	2	0.015	52	3	2	0.005	52	3	2	0.004	52	3	2	0.008
HELIX	3	270	173	142	0.094	632	346	132	0.090	317	164	37	0.040	454	247	96	0.065
BARD	3	92	25	20	0.053	303	155	106	0.218	140	84	60	0.092	141	86	63	0.085
GAUSS	3	8	4	4	0.014	9	5	4	0.003	9	5	4	0.002	9	5	4	0.002
MEYER	3	6	4	2	0.013	7	5	2	0.016	7	5	2	0.002	7	5	2	0.001
GULF	3	3	2	2	0.010	52	4	2	0.007	52	4	2	0.007	52	4	2	0.007
BOX	3	3	2	1	0.001	4	3	1	0.002	4	3	1	0.001	4	3	1	0.001
SING	4	1916	1151	911	0.320	751	410	250	0.104	440	238	152	0.071	891	463	286	0.127
WOOD	4	993	522	405	0.140	682	432	321	0.101	166	92	57	0.027	428	239	152	0.056
KOWOSB	4	3390	1717	1343	0.561	722	438	280	0.129	312	189	125	0.059	368	214	123	0.050
BD	4	155	90	79	0.025	148	85	60	0.022	143	80	50	0.021	166	72	51	0.019
OSB1	5	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
BIGGS	6	5	3	2	0.002	114	66	49	0.022	86	23	20	0.018	63	38	29	0.010
OSB2	11	-	-	-	-	977	56	43	0.148	-	-	-	-	-	-	-	0.016
WATSON	5	5762	3409	2773	2.002	577	355	244	0.189	643	398	272	0.195	449	271	182	0.136
ROSEX	50	258	11	8	0.033	197	108	71	0.037	207	84	48	0.030	263	117	67	0.030
SINGX	4	1916	1151	911	0.330	751	410	250	0.129	288	153	98	0.046	891	463	286	0.113
PEN1	10	941	448	317	0.234	1008	26	21	0.138	9808	202	197	1.308	9813	201	197	3.520
PEN1	50	1672	84	52	0.473	1013	25	21	0.361	9813	201	197	3.541	9811	201	197	7.603
PEN1	100	1672	84	52	0.805	1001	21	20	0.765	9814	201	197	7.570	9851	198	197	13.860
PEN1	200	1672	84	52	1.903	1001	21	20	1.438	9851	198	197	13.664	9851	198	197	44.047
PEN1	500	1672	84	52	6.830	1001	21	20	4.619	9851	198	197	46.588	9851	198	197	83.847
PEN1	800	1672	84	52	7.692	1001	21	20	8.836	9851	198	197	86.425	9851	198	197	116.278
PEN1	1000	1672	84	52	13.206	1001	21	20	11.960	9851	198	197	117.702	9851	198	197	149.359
PEN2	10	1674	89	52	0.177	1002	22	21	0.101	9852	199	198	0.955	9852	199	198	0.926
PEN2	50	1682	95	68	0.214	1022	32	23	0.148	9813	204	198	1.481	9815	205	198	1.497
PEN2	100	571	247	145	0.200	-	-	-	-	1364	481	158	0.349	804	355	167	0.229
PEN2	200	935	80	73	0.358	-	-	-	-	9804	199	198	3.102	9804	199	198	3.015
PEN2	300	1001	21	20	1.833	942	60	59	1.671	9851	198	197	15.841	9851	198	197	13.781
VARDIM	2	25	15	15	0.004	27	13	7	0.010	27	13	7	0.010	29	15	8	0.005
VARDIM	10	143	80	60	0.028	144	75	44	0.071	127	32	17	0.025	103	54	30	0.021
VARDIM	50	987	486	349	0.154	1009	26	22	0.086	9809	202	198	0.790	9809	202	198	0.811
VARDIM	100	154	25	16	0.021	1004	23	22	0.109	9804	199	198	1.050	9804	199	198	1.045
TRIG	10	67	43	42	0.018	88	58	54	0.019	72	46	40	0.014	74	46	39	0.018
TRIG	50	172	52	48	0.059	109	72	65	0.045	94	60	53	0.042	93	57	48	0.040
TRIG	100	178	56	50	0.127	118	80	72	0.135	99	60	49	0.094	132	88	72	0.133
TRIG	200	121	67	56	0.313	106	73	68	0.314	100	63	53	0.274	128	81	66	0.344
TRIG	500	90	59	55	1.962	119	77	67	2.493	130	80	62	2.463	112	70	54	2.289
TRIG	800	110	65	57	6.602	117	80	71	7.806	136	85	67	7.943	244	121	88	12.683

TABLE I - *continued*

p	n	MH				MDY				MHS				MN			
		NF	NG	NI	Tcpu	NF	NG	NI	Tcpu	NF	NG	NI	Tcpu	NF	NG	NI	Tcpu
TRIG	1000	102	62	56	8.152	-	-	-	-	122	77	63	9.657	137	89	69	11.667
TRIG	1200	104	66	58	13.474	511	499	490	92.624	125	78	58	15.305	174	109	78	21.170
TRIG	10	67	43	42	0.018	88	58	54	0.019	72	46	40	0.014	74	46	39	0.018
TRIG	50	172	52	48	0.059	109	72	65	0.045	94	60	53	0.042	93	57	48	0.040
TRIG	100	178	56	50	0.127	118	80	72	0.135	99	60	49	0.094	132	88	72	0.133
TRIG	200	121	67	56	0.313	106	73	68	0.314	100	63	53	0.274	128	81	66	0.344
TRIG	500	90	59	55	1.962	119	77	67	2.493	130	80	62	2.463	112	70	54	2.289
TRIG	800	110	65	57	6.602	117	80	71	7.806	136	85	67	7.943	244	121	88	12.683
TRIG	1000	102	62	56	8.152	-	-	-	-	122	77	63	9.657	137	89	69	11.667
TRIG	1200	104	66	58	13.474	511	499	490	92.624	125	78	58	15.305	174	109	78	21.170
BV	50	637	467	364	0.186	534	488	468	0.173	6374	4661	3627	1.627	6198	4717	3803	1.520
BV	200	641	466	361	0.399	512	492	490	0.349	6366	4655	3635	3.663	6149	4721	3854	3.276
BV	500	631	465	371	2.834	510	492	491	2.891	6397	4646	3605	29.754	6141	4717	3860	25.899
BV	800	638	464	363	6.776	510	493	492	6.804	6348	4662	3655	67.434	6148	4725	3853	51.648
BV	1000	638	465	363	9.225	510	493	492	9.932	6390	4657	3613	92.552	6156	4723	3845	89.167
BV	1200	637	467	366	13.047	510	492	491	13.635	-	-	-	-	6164	4712	3837	128.243
BV	1500	13	9	9	0.418	502	501	500	21.314	-	-	-	-	13	11	10	0.447
BV	2000	6	4	4	0.290	502	501	500	35.810	-	-	-	-	13	11	10	0.807
IE	5	14	7	7	0.002	21	11	10	0.011	15	8	7	0.009	15	8	7	0.003
IE	10	12	6	6	0.005	19	10	9	0.006	13	7	6	0.004	15	8	7	0.006
IE	100	12	6	6	0.105	89	61	60	0.981	16	9	7	0.139	18	10	8	0.164
IE	200	13	7	7	0.449	16	9	8	0.553	14	8	7	0.471	14	8	7	0.505
IE	500	15	8	8	3.111	22	12	10	4.370	16	9	7	3.150	16	9	7	3.311
IE	800	15	8	8	7.899	18	10	8	8.965	18	10	8	8.932	16	9	7	8.397
IE	1000	15	8	8	12.341	16	9	7	12.243	18	10	8	13.905	18	10	8	14.885
IE	2000	18	9	8	55.542	18	10	8	55.506	18	10	8	55.557	16	9	7	52.431
TRID	10	54	27	27	0.011	132	93	91	0.036	65	36	35	0.016	52	27	26	0.013
TRID	50	59	30	30	0.018	135	89	87	0.051	68	36	34	0.020	70	37	35	0.020
TRID	100	59	30	30	0.025	157	106	102	0.079	75	40	36	0.030	72	39	35	0.036
TRID	200	65	32	31	0.046	141	91	87	0.105	72	39	35	0.047	75	40	36	0.057
TRID	500	66	33	32	0.281	154	101	95	0.737	77	40	34	0.302	72	39	33	0.288
TRID	1000	66	33	32	0.897	157	105	100	2.549	75	39	34	1.013	70	38	33	0.941
TRID	1500	69	34	32	1.951	151	98	93	5.008	75	39	34	2.079	78	41	36	2.142
TRID	2000	72	36	35	3.640	150	97	92	8.167	75	39	34	3.521	78	41	36	3.664
BAND	10	62	21	17	0.007	144	91	86	0.030	54	22	17	0.015	50	20	15	0.009
BAND	50	103	38	34	0.031	1019	25	22	0.184	116	29	24	0.030	108	25	20	0.031
BAND	100	95	40	36	0.072	-	-	-	-	9852	200	198	4.566	9852	199	198	13.113
BAND	200	13	7	7	0.417	1002	22	21	1.337	9852	199	198	13.289	249	27	22	2.444
BAND	500	15	8	8	3.140	1005	47	44	9.249	255	30	25	2.598	1910	60	55	41.239
BAND	800	15	8	8	7.149	1002	22	21	21.243	1774	60	52	38.573	401	32	24	14.081
BAND	1000	15	8	8	12.240	983	41	34	32.662	403	35	27	14.239	4	3	1	0.001
LIN	10	3	2	1	0.001	4	3	1	0.008	4	3	1	0.001	4	3	1	0.006
LIN	50	3	2	1	0.003	4	3	1	0.005	4	3	1	0.005	4	3	1	0.007
LIN	100	3	2	1	0.006	4	3	1	0.007	4	3	1	0.007	4	3	1	0.018

TABLE I - *continued*

p	n	MH				MDY				MHS				MN			
		NF	NG	NI	Tcpu	NF	NG	NI	Tcpu	NF	NG	NI	Tcpu	NF	NG	NI	Tcpu
LIN	200	3	2	1	0.015	4	3	1	0.020	4	3	1	0.015	4	3	1	0.073
LIN	500	3	2	1	0.052	4	3	1	0.072	4	3	1	0.068	4	3	1	0.333
LIN	1200	3	2	1	0.243	4	3	1	0.349	4	3	1	0.349	4	3	1	0.505
LIN	1500	3	2	1	0.367	4	3	1	0.535	4	3	1	0.503	4	3	1	0.879
LIN	2000	3	2	1	0.628	4	3	1	0.870	4	3	1	0.880	4	3	1	0.001
LIN1	5	3	2	2	0.001	4	3	1	0.001	4	3	1	0.001	4	3	1	0.001
LIN1	10	3	2	1	0.001	4	3	1	0.001	4	3	1	0.007	6	4	2	0.003
LIN1	50	103	38	34	0.031	6	4	2	0.003	6	4	2	0.003	6	4	2	0.014
LIN1	100	95	40	36	0.072	-	-	-	-	-	-	-	-	-	-	-	-
LIN1	500	642	455	359	19.016	-	-	-	-	-	-	-	-	-	-	-	-
LIN1	800	3	2	2	0.001	-	-	-	-	-	-	-	-	4	3	2	0.001
LIN0	5	3	2	2	0.015	4	3	2	0.008	4	3	2	0.007	4	3	2	0.001
LIN0	10	3	2	2	0.001	4	3	2	0.001	4	3	2	0.001	3	2	1	0.001
LIN0	50	2	1	1	0.001	3	2	1	0.002	27	17	10	0.045	9	5	3	0.018
LIN0	100	14	9	7	0.030	75	49	29	0.114	33	18	6	0.046	349	114	22	1.234

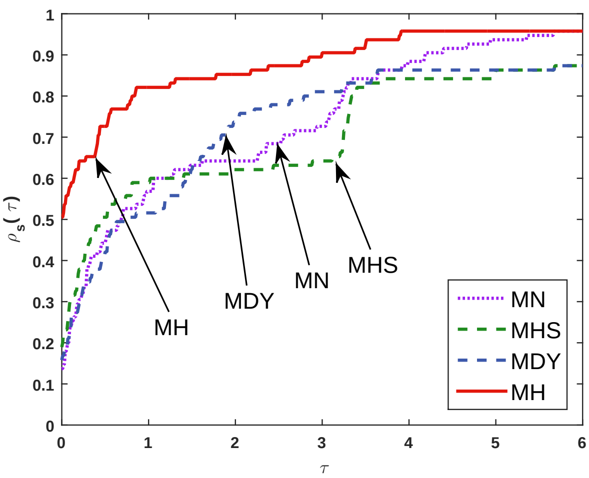

Fig. 4: Performance profiles with respect to CPU time

for the solver $s \in S$ to solve the problem $p \in P$. The comparison between the different solvers are based on the performance ratio defined by

$$r_{p,s} = \frac{t_{p,s}}{min\{t_{p,s} : s \in S\}}.$$

In order to obtain the performance profile for each solver s, define $\rho_s(\tau)$ as the probability of the solver $s \in S$, that is, the performance ratio $r_{p,s}$ is within the factor $\tau \in R$ of the best possible ratio, the distribution function for the performance ratio is defined from

$$\rho_s(\tau) = \frac{1}{n_p} size\{p \in P : \log_2 r_{p,s} \leq \tau\},$$

where size A means the number of elements in the set A and $\tau \geq 0$. If solver s fails to solve a problem p, we set the ratio $r_{p,s}$ to some sufficiently large number.

The corresponding profiles are plotted in Figs. 1, 2, 3 and 4, where Fig. 1, 2, 3 and 4 reveal the efficiency of the above of four algorithms in terms of NF, NG, NI and Tcpu, respectively. Compared with MDY, MHS and MN methods, the proposed method achieves about 60%, 70%, 55% , 50% wins in the number of function evaluations, gradient evaluations, iterations and CPU time. These results demonstrate that the proposed algorithm is promising and performs better on test problems than three existing methods. Therefore, the proposed method has better calculation performance.

References

[1] R. Fletcher and C. M. Reeves, "Function minimization by conjugate gradients, " The Computer Journal, vol. 7, no. 2, pp. 149-154, 1964.
[2] Y. H. Dai and Y. Yuan, "A nonlinear conjugate gradient method with a strong global convergence pcroperty, " SIAM Journal on Optimization, vol. 10, no. 1, pp. 177-182, 1999.
[3] R. Fletcher, Practical Methods of Optimization Vol 1: Unconstrained Optimization, Wiley and Sons, New York, NY, USA, 1987.
[4] M. R. Hestenes and E. Steifel, "Method of conjugate gradients for solving linear equations, " Journal of Research of the National Bureau of Standards, vol. 49, pp. 409-436, 1952.
[5] B. T. Polyak, "The conjugate gradient method in extremal problems," USSR Computational Mathematics and Mathematical Physics, vol. 9, no. 4, pp. 94-112, 1969.
[6] E. Polak and G. Ribiére, "Note sur la convergence deméthodes de directions conjugues," Revue Française D'informatique et de Recherche Opérationnelle. Série Rouge, vol. 3, no. 16, pp. 35-43, 1969.
[7] Y. Liu and C. Storey, "Efcient generalized conjugate gradient algorithms, part 1: theory," Journal of Optimization Teory and Applications, vol. 69, no. 1, pp. 129-137, 1991.
[8] I. A. R. Moghrabi, " A New Preconditioned Conjugate Gradient Method for Optimization," IAENG International Journal of Applied Mathematics, vol. 49, no. 1, pp. 29-36, 2019.

[9] M. Malik, M. Mamat, S. S. Abas, I. M. Sulaiman, and Sukono, “A new coefficient of the conjugate gradient method with the sufficient descent condition and global convergence properties,” Engineering Letters, vol. 28, no. 3, pp. 704-714, 2020.
[10] M. Malik, M. Mamat, S. S. Abas, I. M. Sulaiman, and Sukono, “Performance analysis of new spectral and hybrid conjugate gradient methods for solving unconstrained optimization problems,” IAENG International Journal of Computer Science, vol. 48, no. 1, pp. 66-79, 2021.
[11] M. Malik, A. B. Abubakar, I. M. Sulaiman, M. Mamat, S. S. Abas, and Sukono, “A new three-term conjugate gradient method for unconstrained optimization with applications in portfolio selection and robotic motion control,” IAENG International Journal of Applied Mathematics, vol. 51, no. 3, pp. 471-486, 2021.
[12] Z. X. Wei, H. D. Huang, and Y. R. Tao, “A modified hestenes stiefel conjugate gradient method and its convergence,” Journal of Mathematical Research and Exposition, vol. 30, no. 2, pp. 297-308, 2010.
[13] X. Z. Jiang, J. B. Jian, D. Song, and P. J. Liu, “An improved PolakCRibièreCPolyak conjugate gradient method with an efficient restart direction,” Computational and Applied Mathematics, vol. 40, no. 5, pp. 174-197, 2021.
[14] H. Huang, “A new conjugate gradient method for nonlinear unconstrained optimization problems,” Journal of Henan University(Natural Science), vol. 44, no. 2, pp. 141-145, 2014.
[15] Y. H. Dai and Y. Yuan, “An efficient hybrid conjugate gradient method for unconstrained optimization,” Annals of Operations Research, vol. 103, no. 1, pp. 33-47, 2001.
[16] Y. H. Dai, “A nonmonotone conjugate gradient algorithm for unconstrained optimization,” Journal of Systems Science Complexity, vol. 15, no. 2, pp. 139-145, 2002.
[17] X. Z. Jang, L. Han, and J. B. Jian, “A globally convergent mixed conjugate gradient method with Wolfe line search,” Mathematica Numerica Sinica, vol. 34, no. 1, pp. 103-112, 2012.
[18] J. Jian, L. Han, and X. Jiang, “A hybrid conjugate gradient method with descent property for unconstrained optimization,” Applied Mathematical Modelling, vol. 39, no. 3-4, pp. 1281-1290, 2015.
[19] X. Z. Jiang and J. B. Jian, “A sufficient descent Dai-Yuan type nonlinear conjugate gradient method for unconstrained optimization problems,” Nonlinear Dynamics, vol. 72, no. 1-2, pp. 101-112, 2013.
[20] X. Jiang and J. Jian, “Two modified nonlinear conjugate gradient methods with disturbance fators for unconstrained optimization,” Nonlinear Dynmics, vol. 77, no. 1-2, pp. 387-394, 2014.
[21] J. B. Jian, J. H. Yin, and X. Z. Jiang, “An efficient conjugate gradient method with sufficient descent property.” Mathematica Numerica Sinica, vol. 37, no. 4, pp. 415-424, 2015.
[22] X. Z. Jiang, H. L. Li, M. X. Liu, C. Z. Ban, Y. M. Ma, and Y. W. Ou, “An improved HS conjugate gradient method with sufficient descent property.” Mathematica in Practice and Theory, vol. 50, no. 5, pp. 155-164, 2020.
[23] P. Mtagulwa and P. Kaelo, “An efficient modified PRP-FR hybrid conjugate gradient method for solving unconstrained optimization problems,” Applied Numerical Mathematics, vol. 145, pp. 111-120, 2019.
[24] M. Fang, M. Wang, M. Sun, and R. Chen, “A modified hybrid conjugate gradient method for unconstrained optimization,” Journal of Mathematics, vol. 2021, Article ID 5597863, 9 pages, 2021.
[25] O. J. Adeleke, A. E. Ezugwu, and I. A. Osinuga, “A new family of hybrid conjugate gradient methods for unconstrained optimization,” Statistics, Optimization and Information Computing, vol. 9, no. 2, pp. 399-417, 2021.
[26] I. A. R. Moghrabi, “New two-step conjugate gradient method for unconstrained optimization,” International Journal of Applied Mathematics, vol. 33, no. 5, pp. 853-866, 2020.
[27] G. L. Yuan, Z. X. Wei, and X. W. Lu, “Global convergence of BFGS and PRP methods under a modified weak Wolfe-Powell line search,” Applied Mathematical Modelling, vol. 47, pp. 811-825, 2017.
[28] J. J. Morè, B. S. Garbow, and K. E. Hillstrom, “Testing unconstrained optimization software,” ACM Transactions on Mathematical Software, vol. 7, no. 1, pp. 17-41, 1981.
[29] E. D. Dolan and J. J. Morè, “Benchmarking optimization software with performance profiles,” Mathematical Programming, vol. 91, no. 2, pp. 201-213, 2002.