Hindawi
Journal of Optimization
Volume 2018, Article ID 5057096, 13 pages
https://doi.org/10.1155/2018/5057096

Research Article

A New Modified Three-Term Hestenes–Stiefel Conjugate Gradient Method with Sufficient Descent Property and Its Global Convergence

Bakhtawar Baluch,[1] **Zabidin Salleh**,[2] **and Ahmad Alhawarat**[3]

[1] *School of Informatics and Applied Mathematics, Universiti Malaysia Terengganu, 21030 Kuala Nerus, Terengganu, Malaysia*
[2] *Marine Management Science Research Group, School of Informatics and Applied Mathematics, Universiti Malaysia Terengganu, 21030 Kuala Nerus, Terengganu, Malaysia*
[3] *Department of Mathematics, College of Science, Isra University, Amman, Jordan*

Correspondence should be addressed to Zabidin Salleh; zabidin@umt.edu.my

Received 14 May 2018; Revised 1 August 2018; Accepted 19 August 2018; Published 27 September 2018

Academic Editor: Wlodzimierz Ogryczak

This paper describes a modified three-term Hestenes–Stiefel (HS) method. The original HS method is the earliest conjugate gradient method. Although the HS method achieves global convergence using an exact line search, this is not guaranteed in the case of an inexact line search. In addition, the HS method does not usually satisfy the descent property. Our modified three-term conjugate gradient method possesses a sufficient descent property regardless of the type of line search and guarantees global convergence using the inexact Wolfe–Powell line search. The numerical efficiency of the modified three-term HS method is checked using 75 standard test functions. It is known that three-term conjugate gradient methods are numerically more efficient than two-term conjugate gradient methods. Importantly, this paper quantifies how much better the three-term performance is compared with two-term methods. Thus, in the numerical results, we compare our new modification with an efficient two-term conjugate gradient method. We also compare our modification with a state-of-the-art three-term HS method. Finally, we conclude that our proposed modification is globally convergent and numerically efficient.

1. Introduction

In the field of optimization conjugate gradient methods are a well-known approach for solving large-scale unconstrained optimization problems. The conjugate gradient (CG) methods are simple and have relatively modest storage requirements. This class of methods has a vast number of applications in different areas, especially in the field of engineering [1–3].

Consider the unconstrained optimization problem:

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

where $f : \mathbb{R}^n \longrightarrow \mathbb{R}$ is continuously differentiable and its gradient is $g(x)$. Normally CG methods generate a sequence (x_k) defined by

$$x_{k+1} = x_k + \alpha_k d_k, \quad k = 0, 1, \ldots. \tag{2}$$

In (2), $\alpha_k > 0$ is a general line search and d_k is a search direction given by

$$d_k = \begin{cases} -g_k & \text{if } k = 0, \\ -g_k + \beta_k d_{k-1} & \text{if } k \geq 1, \end{cases} \tag{3}$$

where β_k is a parameter of the CG method. The six pioneering forms of β_k are defined in [4–10].

Line searches may be exact or inexact. Exact line searches are time consuming, computationally expensive, and difficult and require large amounts of storage [11–13]. Thus, inexact line search techniques are often adopted because of their efficiency and global convergence properties. Well-known inexact line search methods include the Wolfe and strong Wolfe techniques, which can be written as

$$f(x_k + \alpha_k d_k) \le f(x_k) + \rho \alpha_k g_k^T d_k, \quad g(x_k + \alpha_k d_k)^T d_k \ge \sigma g_k^T d_k, \tag{4}$$

where $0 < \rho < \sigma < 1$, and

$$f(x_k + \alpha_k d_k) \le f(x_k) + \rho \alpha_k g_k^T d_k, \quad \left| g(x_k + \alpha_k d_k)^T d_k \right| \le \sigma \left| g_k^T d_k \right|. \tag{5}$$

Recently, Alhawarat and Salleh [14], Salleh and Alhawarat [15], and Alhawarat et al. [16, 17] proposed efficient CG and hybrid CG methods that fulfill the required global convergence properties. To improve the existing methods, a three-term CG technique has been introduced. Several different researchers have suggested various modifications to the three-term CG method. For instance, Beale [18] and Nazareth [19] proposed CG methods based on three terms that possess the finite termination property, but these do not perform well in practice [20, 21]. Furthermore, reports by McGuire and Wolfe [22], Deng and Li [23], Zhang et al. [24, 25], Cheng [26], Al-Bayati and Sharif [27], Zhang Xiao and Wei [28], Andrei [29–31], Sugiki et al. [32], Narushima et al. [33], Babaie-Kafaki and Ghanbari [34], Al-Baali et al. [35], Sun and Liu [36], and Baluch et al. [37] discuss the global convergence and numerical results of modified three-term CG methods.

In this paper, a modified three-term Hestenes–Stiefel (HS) method is proposed. The general formula of the HS method [4] is

$$\beta_k^{HS} = \frac{g_k^T y_{k-1}}{d_{k-1}^T y_{k-1}}. \tag{6}$$

This is known to be the first of all the CG parameters. This method ensures the global convergence of the exact line search. A nice property of the HS method is that it satisfies the conjugacy condition, regardless of whether the line search is exact or inexact [38]. However, this method does not satisfy the global convergence property when used with an inexact line search.

In this paper, the method of Zhang et al. [25] is modified with the help of another efficient CG parameter proposed by Wei et al. [39]. An attractive feature of the new three-term HS method is that it satisfies the sufficient descent condition regardless of the line search used. Furthermore, our modification is globally convergent for both convex and nonconvex functions when using an inexact line search. Numerical experiments show that the new modification is more efficient and robust than the MTTHS algorithm proposed by Zhang et al. [25]. The second aspect of this paper is to quantify the improvement of the three-term CG method over two-term approaches. To do this, we consider the efficient two-term CG method [40] given by

$$\beta_k^{DHS} = \frac{\|g_k\|^2 - (\|g_k\| / \|g_{k-1}\|) \left| g_k^T g_{k-1} \right|}{\mu \left| g_k^T d_{k-1} \right| + d_{k-1}^T y_{k-1}}. \tag{7}$$

This DHS [40] method is one of the more efficient CG techniques, as it possesses the sufficient descent property and offers global convergence under Wolfe–Powell line search conditions. The numerical results given by this method are also convincing. Therefore, this two-term CG method is compared with our new modification to quantify the improvement offered by three-term CG methods.

The remainder of this paper is organized as follows. In Section 2, the motivation for and construction of the three-term HS CG method is discussed, and the general form is presented in Algorithm A. Section 3 is divided into two subsections, with Section 3.1 covering the sufficient descent condition and the global convergence properties for convex and nonconvex functions and Section 3.2 presenting detailed numerical results to evaluate the proposed method. Finally, Section 4 concludes this paper.

2. Motivation and Formulas

Zhang et al. [25] proposed the first three-term HS (TTHS) method. This can be written as

$$d_k = \begin{cases} -g_k, & \text{if } k = 0, \\ -g_k + \beta_k^{HS} d_{k-1} - \theta_k^{HS} y_{k-1}, & \text{if } k > 0, \end{cases} \tag{8}$$

$\beta_k^{HS} = g_k^T y_{k-1} / d_{k-1}^T y_{k-1}$ and $\theta_k^{HS} = g_k^T d_{k-1} / d_{k-1}^T y_{k-1}$.

TTHS satisfies the descent property; if an exact line search is used, then it reduces to the original HS method. Further, to guarantee the global convergence properties of the search direction given by (8), a modified (MTTHS) algorithm was introduced with the search direction:

$$d_k = \begin{cases} -g_k, & \text{if } k = 0, \\ -g_k + \beta_k^{MHS} d_{k-1} - \theta_k^{M} z_{k-1}, & \text{if } k > 0, \end{cases} \tag{9}$$

where $\beta_k^{MHS} = g_k^T z_{k-1} / d_{k-1}^T z_{k-1}$, $\theta_k^M = g_k^T d_{k-1} / d_{k-1}^T z_{k-1}$ and $z_k = y_k + t \|g(x_k)\|^r s_k$.

As MTTHS was introduced to prove the global convergence properties of the search direction in (8), the question arises as to why (8) is not used to prove the global convergence properties. Instead of ignoring (8), it should be made efficient and globally convergent. Thus, there is room to modify (8) so as to satisfy the global convergence properties. It is expected that such a modification would outperform the MTTHS algorithm numerically.

Wei et al. [39] proposed an efficient CG parameter given by

$$\beta_k^{VFR} = \frac{\mu_1 \|g_k\|^2}{\mu_2 \left| g_k^T d_{k-1} \right| + \mu_3 \|g_{k-1}\|^2}, \quad \text{for } \mu_1 \epsilon (0, +\infty), \ \mu_2 \epsilon [\mu_1 + \varepsilon_1, +\infty), \ \mu_3 \epsilon (0, +\infty). \tag{10}$$

In this parameter, the term $\mu_2 |g_k^T d_{k-1}|$ plays an important role in satisfying the sufficient descent and global convergence properties. Thus, we take $\mu_2 |g_k^T d_{k-1}|$ from the denominator of the above parameter and use it with (8) to construct a new modified three-term HS method. Hence,

$$\beta_k^{BZA} = \frac{g_k^T (g_k - g_{k-1})}{d_{k-1}^T y_{k-1} + \mu |g_k^T d_{k-1}|}. \tag{11}$$

It is known that the HS method does not converge globally when the objective function is nonconvex. Further, Gilbert and Nocedal [41] showed that the parameter β_k^{HS} must be nonnegative to achieve convergence for nonconvex or nonlinear functions, i.e.,

$$\beta_k^{HS+} = \max\left\{\beta_k^{HS}, 0\right\}. \tag{12}$$

Applying the same technique to our parameter β_k^{BZA} gives

$$\beta_k^{BZA+} = \max\left\{\frac{g_k^T (g_k - g_{k-1})}{d_{k-1}^T y_{k-1} + \mu |g_k^T d_{k-1}|}, 0\right\}, \tag{13}$$

$$\theta_k^{BZA} = \frac{g_k^T d_{k-1}}{d_{k-1}^T y_{k-1} + \mu |g_k^T d_{k-1}|}, \tag{14}$$

where $\mu > 1$. If the line search is exact, then the parameters β_k^{BZA}, β_k^{BZA+}, and θ_k^{BZA} reduce to the original parameters β_k^{HS} [4], β_k^{HS+} [41], and TTHS [25]. The procedure of our proposed three-step CG method is described in Algorithm A.

Algorithm A.

Step 0. Choose an initial point $x_0 \epsilon \mathbb{R}^n$, $\mu > 1$, $0 < \rho < \sigma < 1$, and set $d_0 = -g_0$, $k := 0$.

Step 1. For convergence, if $\|g_k\| \le \varepsilon$ $(= 10^{-6})$, then the algorithm terminates; otherwise, go to step 2.

Step 2. Compute

$$d_k = \begin{cases} -g_k & \text{if } k = 0, \\ -g_k + \beta_k^{BZA} d_{k-1} - \theta_k^{BZA} y_{k-1} & \text{if } k \ge 1. \end{cases} \tag{15}$$

$y_{k-1} = g_k - g_{k-1}$, β_k^{BZA} and θ_k^{BZA} are given in (11) and (14).

Step 3. Determine the step size $\alpha_k > 0$ by the Wolfe line search (4).

Step 4. Compute the new point x_{k+1}.

Step 5. Set $k = k + 1$ and go to step 1.

3. Results and Discussion

This section contains a theoretical discussion and numerical results. The first subsection considers the global convergence properties of our proposed method and the second presents the results from numerical computations.

3.1. Global Convergence Properties

Assumptions

(A1) The level set $\mathbb{R}_0 = \{x | f(x) \le f(x_0)\}$ is bounded.

(A2) In some neighborhood $\mathcal{N}$ of $\mathbb{R}_0$, the gradient $g(x)$ is Lipschitz continuous on an open convex set E that contains $\mathbb{R}_0$, i.e., there exists a positive constant $L > 0$ such that

$$\|g(x_k) - g(x_{k-1})\| \le L \|x_k - x_{k-1}\| \quad \text{for any } x_k, x_{k-1} \in E. \tag{16}$$

Assumptions (A1) and (A2) imply that there exist positive constants γ and b such that

$$\|g(x_k)\| \le \gamma \quad \forall x_k \in \mathbb{R}_0, \tag{17}$$

$$\|x_k - x_{k-1}\| \le b \quad \forall x_k, x_{k-1} \in \mathbb{R}_0. \tag{18}$$

We now prove the sufficient descent condition independent of the line search $g_k^T d_k = -\|g_k\|^2$ and also $\|g_k\| \le \|d_k\|$. From (15), (11), and (14), we can write

$$\begin{aligned} d_k &= -g_k + \beta_k^{BZA} d_{k-1} - \theta_k^{BZA} y_{k-1} \\ g_k^T d_k &= -\|g_k\|^2 + \left(\frac{g_k^T (g_k - g_{k-1}) (g_k^T d_{k-1})}{d_{k-1}^T y_{k-1} + \mu |g_k^T d_{k-1}|}\right) \\ &\quad - \left(\frac{(g_k^T d_{k-1}) g_k^T (g_k - g_{k-1})}{d_{k-1}^T y_{k-1} + \mu |g_k^T d_{k-1}|}\right), \end{aligned} \tag{19}$$

that is,

$$g_k^T d_k = -\|g_k\|^2. \tag{20}$$

Hence, the sufficient descent condition holds regardless of the line search. Now, we prove that

$$\|g_k\| \le \|d_k\|. \tag{21}$$

As we have $g_k^T d_k = -\|g_k\|^2$, taking the modulus on both sides gives

$$|g_k^T d_k| = \left|-\|g_k\|^2\right| = \|g_k\|^2. \tag{22}$$

By the Schwartz inequality, we have

$$|g_k^T d_k| \le \|g_k\| \|d_k\| \tag{23}$$

so

$$\|g_k\|^2 \le \|g_k\| \|d_k\| \tag{24}$$

or

$$\|g_k\| \le \|d_k\|. \tag{25}$$

Hence, we have

$$\begin{aligned} g_k^T d_k &= -\|g_k\|^2, \\ \|g_k\| &\le \|d_k\|. \end{aligned} \tag{26}$$

The HS method is well known for its conjugacy conditions, such as

$$d_k^T y_{k-1} = 0. \tag{27}$$

By [15], CG methods that inherit (27) will be more efficient than other CG parameters that do not inherit this property. Dai and Liao [42] proposed the following conjugacy condition for an inexact line search:

$$d_k^T y_{k-1} = -t\alpha_{k-1} g_k^T d_{k-1}, \quad \text{where } t > 0. \tag{28}$$

Using the exact line search $g_k^T d_{k-1} = 0$, (28) reduces to the conjugacy condition in (27).

Lemma 1 (see [43]). *Suppose there is an initial point x_0 for which Assumptions (A1) and (A2) hold. Now, consider the method in the form of (2), in which d_k is a descent direction and α_k satisfies the Wolfe line search condition (4). Then*

$$\sum_{k=0}^{\infty} \frac{\left(g_k^T d_k\right)^2}{\|d_k\|^2} < +\infty. \tag{29}$$

This is known as Zoutendijk's condition and is used for proving the global convergence of a CG method. This condition together with (26) shows that

$$\sum_{k=0}^{\infty} \frac{\|g_k\|^4}{\|d_k\|^2} < +\infty. \tag{30}$$

Definition 2. The function f is called uniformly convex [36] on $\mathbb{R}^n$ if there exists a positive constant m such that

$$m\|d_k\|^2 \le d^T \nabla^2 f(x_k)\, d \quad \forall x, d \in \mathbb{R}^n. \tag{31}$$

We now show the global convergence of Algorithm A for uniformly convex functions.

Lemma 3. *Let the sequences (x_k) and (d_k) be generated by Algorithm A and suppose that (31) holds. Then,*

$$z_1 \alpha_k \|d_k\|^2 \le -g_k^T d_k, \tag{32}$$

where $z_1 = (1-\rho)^{-1}(m/2)$.

Proof. For details, see Lemma 2.1 of [44]. □

Theorem 4. *Let the conditions in Assumptions (A1) and (A2) hold and the function $f(x)$ be uniformly convex. Then,*

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

Proof. As

$$\begin{aligned} \beta_k^{BZA} &= \frac{g_k^T (g_k - g_{k-1})}{d_{k-1}^T y_{k-1} + \mu |g_k^T d_{k-1}|}, \\ \langle d_{k-1}, y_{k-1}\rangle &= \langle d_{k-1}, g_k - g_{k-1}\rangle \\ &= \langle d_{k-1}, g_k\rangle - \langle d_{k-1}, g_{k-1}\rangle . \end{aligned} \tag{34}$$

Then, using the second Wolfe condition (4) and the sufficient descent condition,

$$\begin{aligned} \langle g(x_k + \alpha_k d_k), d_k\rangle &\ge \sigma_1 \langle g_k, d_k\rangle, \\ \langle g_k, d_k\rangle &= -\|g_k\|^2, \end{aligned} \tag{35}$$

we have

$$\begin{aligned} \langle d_{k-1}, y_{k-1}\rangle &= \langle d_{k-1}, g_k\rangle - \langle d_{k-1}, g_{k-1}\rangle \\ &\ge \sigma_1 \langle g_{k-1}, d_{k-1}\rangle - \langle g_{k-1}, d_{k-1}\rangle \\ &= -(1-\sigma_1)\langle g_{k-1}, d_{k-1}\rangle \\ &= (1-\sigma_1)\|g_{k-1}\|^2 . \end{aligned} \tag{36}$$

From (11), (32), and (36) and Assumption (A2),

$$\begin{aligned} \left|\beta_k^{BZA}\right| &\le \left|\frac{g_k^T (g_k - g_{k-1})}{d_{k-1}^T y_{k-1}}\right| \le \frac{\|g_k\| L \|x_k - x_{k-1}\|}{(1-\sigma_1)\|g_{k-1}\|^2} \\ &= \frac{\|g_k\| L \|s_{k-1}\|}{(1-\sigma_1)\left(-g_{k-1}^T d_{k-1}\right)} \\ &\le \frac{L\|g_k\| \alpha_{k-1}\|d_{k-1}\|}{(1-\sigma_1) z_1 \alpha_{k-1}\|d_{k-1}\|^2} \\ &= \frac{L\|g_k\|}{(1-\sigma_1) z_1 \|d_{k-1}\|}. \end{aligned} \tag{37}$$

Let us suppose that $(1-\sigma_1)z_1 = z_2$, where $0 < \sigma_1 < 1$ and $z_1 > 0$ so that $z_2 > 0$. Thus,

$$\left|\beta_k^{BZA}\right| \le \frac{L\|g_k\|}{z_2\|d_{k-1}\|} \Longrightarrow \left|\beta_k^{BZA}\right| \|d_{k-1}\| \le \frac{L\|g_k\|}{z_2}. \tag{38}$$

Now,

$$\begin{aligned} \left|\theta_k^{BZA}\right| &\le \left|\frac{g_k^T d_{k-1}}{d_{k-1}^T y_{k-1}}\right| \le \frac{\|g_k\|\|d_{k-1}\|}{(1-\sigma_1) z_1 \alpha_{k-1}\|d_{k-1}\|^2} \\ \left|\theta_k^{BZA}\right| \|y_{k-1}\| &\le \frac{\|g_k\|\|d_{k-1}\|\|y_{k-1}\|}{(1-\sigma_1) z_1 \alpha_{k-1}\|d_{k-1}\|^2} \\ &\le \frac{\|g_k\|\|d_{k-1}\| L\|s_{k-1}\|}{z_2 \alpha_{k-1}\|d_{k-1}\|^2} \\ &= \frac{\|g_k\|\|d_{k-1}\| L\alpha_{k-1}\|d_{k-1}\|}{z_2 \alpha_{k-1}\|d_{k-1}\|^2} = \frac{L\|g_k\|}{z_2}. \end{aligned} \tag{39}$$

Combining (38) and (39) with (15), we obtain

$$\begin{aligned} \|d_k\| &\le \|g_k\| + \left|\beta_k^{BZA}\right| \|d_{k-1}\| + \left|\theta_k^{BZA}\right| \|y_{k-1}\| \\ &\le \|g_k\| + \frac{L\|g_k\|}{z_2} + \frac{L\|g_k\|}{z_2} = \|g_k\| + \frac{2L\|g_k\|}{z_2} \\ &\le \left(1 + \frac{2L}{z_2}\right)\|g_k\| . \end{aligned} \tag{40}$$

Now, let $\sqrt{C} = 1 + 2L/z_2$ so that

$$\|d_k\| \leq \sqrt{C}\,\|g_k\|, \tag{41}$$

and we get $\|d_k\|^2 \leq C\|g_k\|^2$. This implies that

$$\begin{aligned} \frac{1}{C\|g_k\|^2} &\leq \frac{1}{\|d_k\|^2} \\ \frac{\|g_k\|^4}{\|g_k\|^2} &\leq \frac{C\|g_k\|^4}{\|d_k\|^2}. \end{aligned} \tag{42}$$

Hence, by (30), we have

$$\lim_{k\longrightarrow\infty}\|g_k\|^2 \leq C \lim_{k\longrightarrow\infty}\frac{\|g_k\|^4}{\|d_k\|^2} = 0. \tag{43}$$

□

We are now going to prove the global convergence of Algorithm A for nonconvex functions.

Lemma 5. *Suppose that Assumptions (A1) and (A2) hold. Let the sequence (x_k) be generated by Algorithm A. If there exists a constant $\epsilon > 0$ such that $\|g_k\| \geq \epsilon$ for every $k \geq 0$, then*

$$\sum_{k=0}^{\infty}\|u_{k+1} - u_k\|^2 < +\infty, \tag{44}$$

where $u_k = d_k/\|d_k\|$.

Proof. As $\|g_k\| \leq \|d_k\|$ and $g_k^T d_k = -\|g_k\|^2$, and also $\|g_k\| \geq \epsilon$ for all k, then $\|d_k\| > 0$ for all k. Hence, u_k is well defined. If

$$\begin{aligned} r_k &= -\frac{\left(1 + \theta_k^{BZA} {g_k}^T y_{k-1}/\|g_k\|^2\right) g_k}{\|d_k\|}, \\ \delta_k &= \beta_k^{BZA+}\frac{\|d_{k-1}\|}{\|d_k\|}, \end{aligned} \tag{45}$$

then $u_k = r_k + \delta_k u_{k-1}$, where u_k and u_{k-1} are unit vectors. Therefore,

$$\|r_k\| = \|\delta_k u_k - u_{k-1}\| = \|u_k - \delta_k u_{k-1}\|. \tag{46}$$

As $\delta_k \geq 0$,

$$\begin{aligned} \|u_k - u_{k-1}\| &\leq \|(1+\delta_k)(u_k - u_{k-1})\| \\ &\leq \|u_k - \delta_k u_{k-1}\| + \|\delta_k u_k - u_{k-1}\| \\ &= 2\|r_k\|. \end{aligned} \tag{47}$$

Now, from Assumption (A2), (14), and (18),

$$\begin{aligned} \left|\theta_k^{BZA}\right| \frac{\|g_k^T\|\,\|y_{k-1}\|}{\|g_k\|^2} &\leq \left|\frac{g_k^T d_{k-1}}{\mu\left|g_k^T d_{k-1}\right|}\right| \frac{\|g_k^T\|\,\|y_{k-1}\|}{\|g_k\|^2} \\ &\leq \frac{L\|x_k - x_{k-1}\|}{\mu\|g_k\|} \leq \frac{Lb}{\mu\epsilon}. \end{aligned} \tag{48}$$

From (17), (18), and (48), there exists a constant $N_1 \geq 0$ such that

$$\begin{aligned} &\left\|-\left(1 + \frac{\theta_k^{BZA}{g_k}^T y_{k-1}}{\|g_k\|^2}\right) g_k\right\| \\ &\leq \|g_k\| + \left(\left|\theta_k^{BZA}\right| \frac{\|g_k^T\|\,\|y_{k-1}\|}{\|g_k\|^2}\right)\|g_k\| \leq \gamma + \frac{Lb}{\epsilon\mu}\gamma \\ &= N_1. \end{aligned} \tag{49}$$

From (30) and (49), we obtain

$$\begin{aligned} \sum_{k=0}^{\infty}\|r_k\|^2 &\leq \sum_{k=0}^{\infty}\frac{N_1^2}{\|d_k\|^2} \leq \sum_{k=0}^{\infty}\frac{N_1^2}{\|g_k\|^4}\frac{\|g_k\|^4}{\|d_k\|^2} \\ &\leq \frac{N_1^2}{\epsilon^4}\sum_{k=0}^{\infty}\frac{\|g_k\|^4}{\|d_k\|^2} < +\infty. \end{aligned} \tag{50}$$

Combining this with (44) completes the proof. □

Theorem 6. *Let Assumptions (A1) and (A2) hold. Then, the sequence (x_k) generated by Algorithm A satisfies*

$$\lim_{k\longrightarrow\infty}\inf\|g_k\| = 0. \tag{51}$$

Proof. Suppose that $\lim_{k\longrightarrow\infty}\inf\|g_k\| \neq 0$. Then, there exists a constant $\epsilon > 0$ such that $\|g_k\| \geq \epsilon\ \forall k \geq 0$.

The proof has two parts.

Part 1. See Theorem 2.2, step 1 in [36].

Part 2. From (15) and (49), we have

$$\begin{aligned} &\|d_k\|^2 \\ &\leq \left(\left|\beta_k^{BZA+}\right|\|d_{k-1}\| + \left\|-\left(1 + \frac{\theta_k^{BZA}g_k^T y_{k-1}}{\|g_k\|^2}\right)g_k\right\|\right)^2 \\ &\leq \left(\left|\beta_k^{BZA+}\right|\|d_{k-1}\| + N_1\right)^2 \\ &\leq 2\left(\left|\beta_k^{BZA+}\right|\|d_{k-1}\|\right)^2 + 2N_1^2 \\ &\leq 2\left(\frac{g_k^T(g_k - g_{k-1})}{\varepsilon}\right)^2\|d_{k-1}\|^2 + 2N_1^2 \\ &\leq 2\left(\frac{\|g_k\|\,L\,\|s_{k-1}\|}{\varepsilon}\right)^2\|d_{k-1}\|^2 + 2N_1^2 \\ &\leq 2\frac{\gamma^2 L^2\|s_{k-1}\|^2}{\varepsilon^2}\|d_{k-1}\|^2 + 2N_1^2. \end{aligned} \tag{52}$$

In the beginning of the proof, we suppose that $\lim_{k\longrightarrow\infty}\inf\|g_k\| \neq 0$. Then, there exist a positive constant ϵ and some $\gamma > 0$ such that $\|g_k\| > \gamma > 0$. Thus,

$$\sum_{k=0}^{\infty}\frac{\left(g_k^T d_k\right)^2}{\|d_k\|^2} > +\infty \Longrightarrow \frac{1}{\|d_k\|^2} > +\infty, \tag{53}$$

which contradicts Assumption (A2), (30), and (52). Therefore,

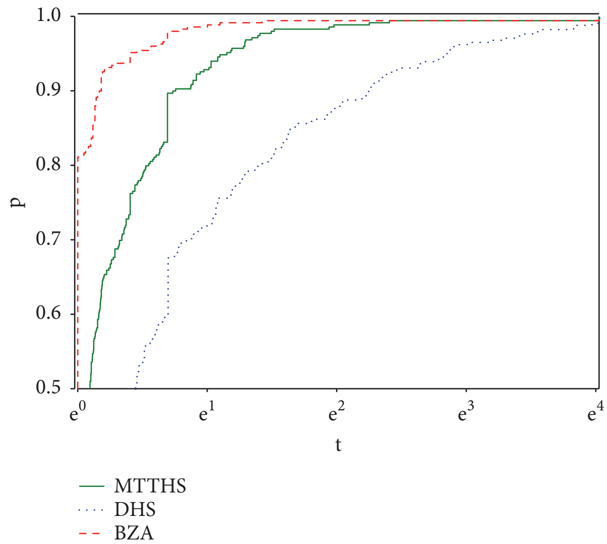

Figure 1: Performance profiles based on number of iterations.

$$\lim_{k\longrightarrow\infty} \inf \left\| g_k \right\| = 0. \tag{54}$$

□

3.2. Numerical Discussion. We now report the results of several numerical experiments. Zhang et al. [25] demonstrated the superior numerical efficiency of the MTTHS algorithm with respect to PRP+ [41], CG_DESCENT [45], and L-BFGS [46] using the Wolfe line search, while Dai and Wen [40] reported the numerical efficiency of the DHS method. Thus, we compare the efficient three-term HS method proposed in this paper (named the Bakhtawar–Zabidin–Ahmad method, BZA) with MTTHS [25] and DHS [40]. The BZA method was implemented using the Wolfe–Powell line search (4) with $\rho = 0.1$, $\sigma = 0.5$, and $\mu = 2$.

All codes were written in MATLAB 7.1 and run on an Intel Core i5 system with 8.0 GB RAM and a 2.60 GHz processor. Table 1 lists the numerical results given by BZA, MTTHS, and DHS for a number of test functions. In the Table 1, NI/CT/GE/FE represents number of iterations, CPU time, number of gradient evaluations and number of function evaluations.

According to Moré et al. [47], the efficiency of any method can be determined by its performance on a number of test functions. The number of test functions should not be too large or too small, with 75 considered ideal for testing the efficiency of any method. The test functions in Table 1 were taken from Andrei's test function collection [48] with standard initial points and dimensions ranging from 2 to 10000.

If the solution had not converged after 500 seconds, the program was terminated. Generally, convergence was achieved within this time limit; functions for which the time limit was exceeded are denoted by "F" for Fail in Table 1.

The Sigma plotting software was used to graph the data. We adopt the performance profiles given by Dolan and Moré [49]. Thus, MTTHS, DHS, and BZA are compared in terms of NI/CT/GE/FE in Figures 1–4. For each method, we plotted

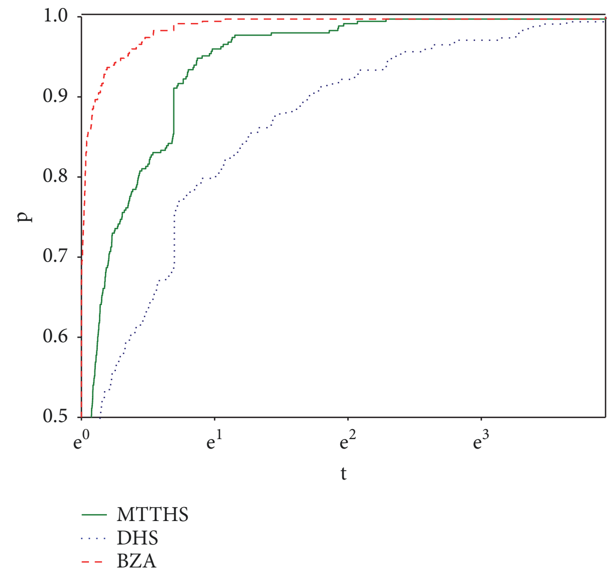

Figure 2: Performance profiles based on CPU time.

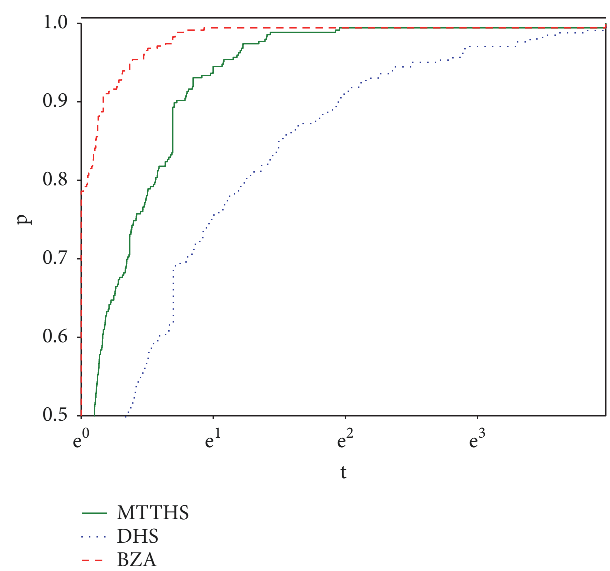

Figure 3: Performance profiles based on gradient evaluation.

the fraction P of problems that were solved correctly within a factor t of the best time. In the figures, the uppermost curve is the method that solves the most problems within a factor t of the best time. From Table 1 and Figures 1–4, the BZA method outperforms the MTTHS algorithm and DHS method in terms of NI, CT, GE, and FE.

The BZA method solves around 99.5% of the problems, and the performance of BZA is 85% better than that of DHS and 77% better than that of MTTHS. We can also conclude that, on average, three-term conjugate gradient methods are 85% better than two-term conjugate gradient methods (DHS).

Table 1: List of test problem functions.

Problem	n	MTTHS NI/CT/GE/FE	DHS NI/CT/GE/FE	BZA NI/CT/GE/FE
Extended	2	1323/9.3448/4645/3321	1267/10.1753/4756/3488	364/3.1079/1365/1000
Trigonometric -1	50	4533/28.4686/15849/11315	3771/25.5829/14146/10374	2376/18.4341/8910/6533
function	500	11125/67.51/38848/27722	8163/55.0526/30616/22452	5904/46.5577/22140/16235
HIMMELBH(Cute)	2	9/0.4680/77/67	9/0.4538/77/67	9/0.4471/77/67
	50	9/0.4572/77/67	9/0.4574/77/67	9/0.4620/77/67
	5000	9/2.0529/77/67	9/2.0198/77/67	9/1.9930/77/67
Power function	2	3/0.3968/13/9	2/0.3881/9/6	2/0.3884/9/6
	50	212/2.0540/849/636	3431/24.0407/13725/10293	65/0.7751/261/195
	1000	4612/33.5261/18449/13836	F/F/F/F	1565/11.567/6261/4695
DENSCHNF	2	11/0.6773/64/52	12/0.5399/67/54	8/0.4719/50/41
	50	14/0.5959/78/63	14/0.5260/75/60	9/0.4739/54/44
	5000	10/1.7879/56/45	14/2.1633/75/60	9/1.6783/54/44
Sum Squares	2	3/0.4203/13/9	2/0.3861/9/6	2/0.3966/9/6
function	50	47/0.7252/189/141	39/0.6318/157/117	39/0.6162/157/117
	1000	229/2.4542/917/687	1333/11.3244/5333/3999	191/1.9552/765/573
TRIDIA(Cute)	2	3/0.4056/13/9	2/0.3980/9/6	2/0.4024/9/6
	50	100/1.6029/401/300	193/1.7752/773/579	56/0.7390/225/168
	1000	604/5.2318/2417/1812	16019/122.48/64077/48057	349/3.22771397/1047
SINQUAD(Cute)	2	25/0.4804/102/76	1734/12.6402/6076/4341	25/0.5218/101/75
	5000	F/F/F/F	F/F/F/F	411/198.6198/2837/2425
	10000	F/F/F/F	F/F/F/F	417/493.6830/2650/2232
Generalized	50	23/0.5659/101/77	25/0.5348/109/83	21/0.5187/94/72
Quartic GQ2	100	23/0.5588/103/79	31/0.5938/133/101	22/0.5233/96/73
function.	5000	26/2.9409/116/89	23/2.6517/103/79	25/2.7548/110/84
Generalized	50	50/0.7767/248/197	43/0.6750/192/148	42/0.6934/198/155
Triagonal-2	500	43/1.5284/206/162	46/1.4306/209/162	38/1.4029/182/143
function	1000	45/2.1144/214/168	47/2.1176/214/166	39/1.8380/181/141
Extended	2	8/0.4315/32/23	9/0.4256/37/27	6/0.4158/25/18
Trigonometric -2	50	33/0.6514/148/114	26/0.5848/124/97	22/0.5586/106/83
function	7000	85/63.5473/626/540	71/44.8822/440/368	33/25.6757/251/217
LIARWHD	100	20/0.5798/136/115	16/0.5065/95/78	14/0.4865/84/69
	5000	50/8.8611/390/339	53/8.7426/384/330	24/4.4628/189/164
	10000	33/14.6362/245/211	53/19.3591/331/277	23/9.7171/165/141
Generalized	2	6/0.4058/27/20	6/0.4127/27/20	5/0.4008/23/17
Quartic GQ1	6000	11/1.8086/47/35	10/1.6695/43/32	10/1.7246/43/32
function	10000	11/3.4464/47/35	10/3.0683/43/32	10/3.0317/43/32
Extended QP2	2	17/0.8111/98/80	21/0.6076/125/103	16/0.5702/111/94
Quadratic Penalty	5000	226/56.4248/2247/2020	F/F/F/F	82/25.7327/1018/935
function	10000	137/95.8189/1457/1319	F/F/F/F	82/67.1490/1024/941
Extended Maratos	2	20/0.6479/131/110	43/0.8956/225/181	30/0.8305/193/162
function	1000	26/0.8290/176/149	F/F/F/F	30/0.8300/193/162
	5000	F/F/F/F	45/5.8526/233/187	32/6.0318/243/210
DENSCHNC	2	6/0.4508/36/29	10/0.4624/51/40	6/0.4486/34/27
	50	7/0.4499/38/30	10/0.4590/51/40	6/0.4696/34/27
	600	8/0.5625/44/35	15/0.6240/130/114	7/0.5445/38/30
DIXON3DQ (Cute)	2	3/0.4835/13/9	2/0.3904/9/6	2/0.3878/9/6
	20	106/1.2286/424/317	262/2.4917/1061/798	90/1.0709/360/269
	600	2266/18.9637/9064/6797	F/F/F/F	2185/18.4756/8740/6554

Table 1: Continued.

Problem	n	MTTHS NI/CT/GE/FE	DHS NI/CT/GE/FE	BZA NI/CT/GE/FE
EDENSCH function	50	3094/15.3448/9288/6193	3094/16.4610/9288/6193	3089/15.3239/9273/6183
	1000	3094/17.6514/9288/6193	3094/18.4918/9288/6193	3089/17.6263/9273/6183
	5000	3094/198.8550/9288/6193	3094/199.0048/9288/6193	3089/198.2240/9273/6183
Diagonal 2 function	50	749/6.8385/2626/1876	495/3.0663/1488/992	490/3.3271/1473/982
	1000	16517/126.4914/59593/43075	8569/53.8256/25711/17141	8563/53.3166/25693/17129
	10000	F/F/F/F	16484/283.6569/49456/32971	16478/279.44/49438/32959
DQDRTIC(Cute)	50	12/0.4358/49/36	5/0.4070/21/15	5/0.3940/21/15
	5000	14/1.6324/57/42	5/0.7966/21/15	5/0.8084/21/15
	10000	14/3.7862/57/42	5/1.6205/21/15	5/1.5979/21/15
Perturbed Quadratic function	50	43/0.6470/173/129	38/0.5837/153/114	38/0.6313/153/114
	1000	207/2.1710/829/621	1397/10.8367/5589/4196	187/1.9258/749/561
	5000	481/44.4081/1925/1443	5125/435.0973/20501/15375	425/39.4768/1701/1275
Diagonal 4 function	50	4/0.3919/17/12	2/0.4015/9/6	2/0.3963/9/6
	1000	5/0.4493/21/15	2/0.4005/9/6	2/0.4039/9/6
	5000	5/0.8522/21/15	2/0.5598/9/6	2/0.5698/9/6
Extended Beale function	50	15/0.4685/74/58	68/0.8247/281/212	15/0.4890/72/56
	100	16/0.4739/74/57	68/0.8245/281/212	15/0.4855/72/56
	500	16/0.5450/74/57	68/0.8840/281/212	15/0.5050/72/56
DENSCHNA	50	9/0.4336/44/34	24/0.5609/102/77	9/0.4462/44/34
	1000	10/0.5912/48/37	26/0.7584/110/83	9/0.5909/44/34
	5000	10/1.6387/46/35	26/3.0388/110/83	9/1.4694/44/34
STAIRCASE S1	20	79/0.9390/316/236	202/1.7976/818/615	72/0.8546/290/217
	50	215/2.0539/860/644	1117/8.6778/4503/3385	183/1.6893/732/548
	100	426/3.4107/1705/1278	3896/29.4293/15737/11840	372/3.0840/1490/1117
NONDQUAR	50	28674/216.605/114767/86092	F/F/F/F	14756/104.86/59098/44341
	100	32283/244.026/129157/96873	F/F/F/F	17900/127.16/71670/53769
	500	F/F/F/F	F/F/F/F	32748/241.1/131043/98294
Extended Wood	500	128/1.5299/605/476	942/8.5789/4143/3200	96/1.4535/551/454
	1000	146/1.7443/676/529	946/8.8946/4100/3153	101/1.7151/606/504
	10000	213/57.8431/979/765	956/249.5361/4202/3245	89/31.8805/545/455
Extended Penalty	20	33/0.6528/144/110	37/0.7031/158/120	22/0.5670/98/75
	300	F/F/F/F	F/F/F/F	64/1.2248/365/300
	600	F/F/F/F	F/F/F/F	28/0.7428/172/143
Sphere function	2	1/0.4750/5/3	1/0.3777/5/3	1/0.3902/5/3
	1000	1/0.3904/5/3	1/0.3877/5/3	1/0.3860/5/3
	5000	1/0.4687/5/3	1/0.4576/5/3	1/0.4616/5/3
ARGLINB(Cute)	2	1/0.3913/5/3	1/0.3903/5/3	1/0.3840/5/3
	50	1/0.3868/5/3	1/0.3941/5/3	1/0.3832/5/3
	500	2/0.4279/9/6	2/0.4266/9/6	2/0.4246/9/6
Extended White and Holst function	2	28/0.7668/192/163	69/1.2885/354/284	21/0.6586/136/114
	500	29/0.9337/213/183	102/1.5134/488/385	21/0.6586/136/114
	5000	23/4.6845/168/144	125/14.3950/577/451	21/3.8322/136/114
Extended Hiebert	50	2040/33.6457/16304/14263	F/F/F/F	838/21.7040/9977/9138
	200	F/F/F/F	F/F/F/F	901/23.007/10513/9611
	1000	F/F/F/F	F/F/F/F	F/F/F/F

Table 1: Continued.

Problem	n	MTTHS NI/CT/GE/FE	DHS NI/CT/GE/FE	BZA NI/CT/GE/FE
Quadratic QF1	50	44/0.6479/177/132	38/0.6068/153/114	38/0.6306/153/114
	500	147/1.4565/589/441	574/4.7399/2297/1722	131/1.3814/525/393
	10000	744/179.6695/2977/2232	F/F/F/F	606/146.3228/2425/1818
Quartic	50	11579/66.6569/34740/23160	11579/66.7991/34740/23160	11574/64.593/34725/23150
	500	24975/148.8391/74928/49952	24975/146.0726/74928/49952	24968/142.57/74907/49938
	1000	31473/201.6973/94422/62948	31473/218.8475/94422/62948	31466/191.54/94401/62934
Shallow function	1000	10/0.5712/45/34	48/0.7436/200/151	10/0.5557/45/34
	5000	11/1.4925/49/37	48/4.9218/200/151	10/1.4117/45/34
	10000	12/3.6303/53/40	50/13.2094/209/158	10/3.1569/45/34
VARDIM	50	28/1.1175/322/293	143/4.0110/1604/1460	16/0.8685/221/204
	100	20/1.0429/286/265	293/5.9711/2604/2310	17/0.9508/250/232
	500	41/2.2545/736/694	1557/25.9389/11724/10166	34/1.8745/645/610
DIXMAANA	6000	8/1.8623/35/26	8/1.8913/35/26	8/1.8504/35/26
	6015	8/1.9276/35/26	8/1.8546/35/26	8/1.9200/35/26
	6030	8/1.9301/35/26	8/1.9700/35/26	8/1.8819/35/26
DIXMAANB	9	7/0.4044/30/22	7/0.4141/30/22	7/0.4036/30/22
	300	7/0.4550/30/22	7/0.4776/30/22	8/0.4415/34/25
	6000	8/1.8966/34/25	8/1.8364/34/25	9/2.0362/38/28
DIXMAANC	9	6/0.4033/29/22	6/0.4191/29/22	6/0.4168/29/22
	300	7/0.4701/33/25	6/0.4721/29/22	7/0.4707/33/25
	6000	7/2.0093/33/25	7/2.0148/33/25	8/2.1736/37/28
DIXMAAND	90	7/0.4450/34/26	8/0.4333/38/29	8/0.4263/38/29
	300	8/0.5272/38/29	8/0.5406/38/29	8/0.5418/38/29
	6000	9/2.4166/42/32	7/2.0165/34/26	7/2.0556/34/26
DIXMAANE	9	19/0.5080/88/68	23/0.5720/107/83	18/0.4832/84/65
	300	84/1.5278/421/336	1211/7.7407/3707/2495	84/2.1712/420/335
	6000	331/81.2267/1655/1323	F/F/F/F	333/83.0952/1685/1351
EG2	2	5/0.4912/25/19	8/0.4128/40/31	4/0.4139/20/15
	20	160/1.4413/642/481	1958/14.8104/8106/6147	110/1.0755/447/336
	50	F/F/F/F	48/0.7452/228/179	656/5.3312/2773/2116
EG3	20	21/0.5445/88/66	25/0.5611/102/76	14/0.4851/60/45
	50	25/0.5965/110/84	43/0.7618/207/163	25/0.5236/69/53
	100	F/F/F/F	F/F/F/F	20/0.5290/88/67
Fletcher function	50	24/0.5879/126/101	23/0.5870/123/99	22/0.5687/118/95
	6000	27/5.2874/161/133	28/5.4497/168/139	26/4.7950/152/125
	10000	27/10.4312/161/133	27/10.6308/163/135	26/9.8879/155/128
Extended Himmelblau function	50	8/0.4229/37/28	11/0.4510/49/37	9/0.4370/41/31
	1000	8/0.4706/39/30	12/0. 5342/53/40	9/0.4775/41/31
	5000	9/1.4066/43/33	13/1.6584/57/43	9/1.3377/41/31
Extended Freudenstein and Roth function	2	11/0.5889/75/63	33/0.6806/163/129	10/0.4929/69/58
	50	F/F/F/F	36/0.7871/230/193	10/0.4700/69/58
	200	16/0.5442/102/85	37/0.8501/241/203	11/0.4993/73/61
Dixon and Price function	20	132/1.6580/615/482	333/3.2018/1409/1075	131/1.5682/603/471
	50	185/2.5157/1034/848	478/4.7648/2199/1720	165/2.2048/900/734
	100	523/6.5773/2938/2414	1243/11.1703/5672/4428	439/5.4568/2422/1982
Raydan 1 function	2	106/1.3363/319/212	105/0.8502/316/210	99/0.8493/298/198
	50	59/0.8854/237/177	88/1.0277/364/275	58/0.7714/233/174
	100	70/0.9532/282/211	155/1.5252/625/469	70/0.8892/282/211
Raydan 2 function	2	2/0.4129/9/6	2/0.3995/9/6	3/0.3971/13/9
	50	2/0.3991/9/6	2/0.3981/9/6	3/0.4176/13/9
	100	2/0.4002/9/6	2/0.4123/9/6	3/0.3906/13/9

Table 1: Continued.

Problem	n	MTTHS NI/CT/GE/FE	DHS NI/CT/GE/FE	BZA NI/CT/GE/FE
NONDIA(SHANO-78)	500	9/0.5483/63/53	11/0.4937/70/58	7/0.4938/55/47
	6000	14/2.7799/82/67	46/6.5297/212/165	9/2.3946/70/60
	10000	16/6.3412/96/79	74/21.4446/339/264	15/6.2113/92/76
Extended Block Diagonal BD1 function	500	30/1.0226/52000/44250	32/0.9386/32250/24000	39/1.1887/44000/34000
	1000	39/3.6068/195000/175000	33/1.3795/66500/49500	39/1.6724/88500/68500
	10000	38/22.3738/1070000/875000	35/14.5725/705000/525000	43/20.1088/965000/745000
SINCOS	2000	F/F/F/F	F/F/F/F	8/0.6960/39/30
	5000	F/F/F/F	F/F/F/F	8/1.2553/39/30
	10000	F/F/F/F	F/F/F/F	8/2.7535/39/30
DIXMAANH	90	58/0.8867/290/231	388/2.5254/1230/841	57/1.0443/286/228
	300	81/1.5051/408/326	1264/7.8924/3872/2607	81/1.3140/408/326
	600	126/4.4107/630/503	2407/40.8602/7294/4886	119/4.0122/596/476
Quadratic QF2 function	50	77/0.9476/330/252	107/1.1580/454/346	68/0.8775/293/224
	200	183/1.7984/787/603	337/2.9913/1419/1081	139/1.5/610/470
	2000	1150/34.4704/4862/3711	2903/85.3362/12110/9206	540/17.4638/2404/1863
Tridiagonal double Bordered	20	108/1.1547/438/329	1915/14.1311/7667/5751	101/1.1136/410/308
	50	407/3.3266/1633/1225	11157/78.2325/44633/33475	355/2.9421/1425/1069
	500	6278/48.1130/25119/18840	13470/103.5476/53887/40416	4595/35.3993/18387/13791
Generalized Triagonal1 function	2	1323/8.3265/4645/3321	1267/8.4042/4756/3488	364/2.6867/1365/1000
	50	25/0.5255/103/77	28/0.504/115/86	24/0.5140/99/74
	100	24/0.5630/99/74	27/0.6205/111/83	23/0.5242/95/71
Extended QP2 Quadratic penalty function	50	9/0.4269/49/39	15/0.4503/71/55	9/0.4190/49/39
	200	13/0.4682/70/56	14/0.6931/74/59	12/0.4620/66/53
	3000	F/F/F/F	F/F/F/F	17/1.5340/118/100
Extended DENSCHNB function	2	5/0.4089/22/16	6/0.4254/26/19	5/0.4095/22/16
	500	5/0.4222/22/16	7/0.4313/30/22	6/0.4165/26/19
	10000	5/1.7240/22/16	7/2.1577/30/22	6/1.9461/26/19
Extended three-Exponential terms	2	7/0.4187/30/22	13/0.4386/54/40	7/0.417/30/22
	50	7/0.4211/750/550	14/0.4492/1450/1075	7/0.4255/750/550
	100	7/0.4378/1500/1100	14/0.4705/2900/2150	7/0.4183/1500/1100
DIXMAANF	9	23/0.6308/109/85	23/0.5828/109/85	19/0.5129/90/70
	90	51/0.8509/256/204	386/2.4637/1229/842	51/1.1421/256/204
	300	90/1.4444/451/360	1260/7.8258/3871/2610	90/1.5578/451/360
DIXMAANG	9	19/0.5318/95/75	26/0.5670/127/100	21/0.5403/103/81
	90	83/1.2240/424/340	379/2.4949/1221/841	56/0.8767/280/223
	300	137/1.8866/643/505	1241/14.5812/3834/2592	569/5.5495/2477/1907
Extended Rosenbrock function	2	25/0.7100/159/133	24/0.9013/120/95	27/0.7803/163/135
	1000	14/0.8076/82/67	26/0.6537/128/101	28/0.7818/167/138
	5000	19/2.8693/109/89	27/3.2102/132/104	30/4.4233/175/144
ARWHEAD	500	F/F/F/F	F/F/F/F	10/0.4429/54/43
	3000	F/F/F/F	F/F/F/F	5/0.7593/38/32
	8000	F/F/F/F	F/F/F/F	7/2.5096/49/41
Hager function	2	4/0.4001/17/12	6/0.4188/25/18	4/0.4008/17/12
	50	20/0.4789/82/61	24/0.5101/98/73	20/0.4738/82/61
	100	24/0.5275/98/73	F/F/F/F	24/0.5382/111/86
Extended Powell function	1000	4008/108.2/4032500/3030250	F/F/F/F	361/10.5/384000/293500
	3000	1026/86.1/3124500/2354250	F/F/F/F	108/10.8/379500/297750
	5000	2213/330/11330000/8562500	F/F/F/F	306/48/1652500/1268750
BIGGSB1 function	2	1/0.3861/5/3	1/0.3866/5/3	1/0.3895/5/3
	20	53/0.7214/213/159	20/0.4869/80/59	20/0.4764/80/59
	50	201/1.9444/804/602	867/7.1960/3500/2632	50/0.7281/200/149

Table 1: Continued.

Problem	n	MTTHS NI/CT/GE/FE	DHS NI/CT/GE/FE	BZA NI/CT/GE/FE
Extended Cliff	100	11657/87/2075150/1492250	39907/329/7892000/5896600	1674/12/293750/210000
	5000	F/F/F/F	F/F/F/F	1009/132/8857500/633250
ENGVAL8	2	7/0.4675/33/25	10/0.4709/45/34	7/0.4756/36/28
	20	22/0.5657/99/76	32/0.6306/138/105	20/0.5382/90/69
Trecanni function	2	6/0.4146/26/19	6/0.4212/26/19	4/0.3982/18/13
GENROSEN-2	2	25/0.8096/159/133	24/0.5864/120/95	27/0.6770/163/135
Generalized Quartic function	2	6/0.4068/27/20	6/0.4082/27/20	5/0.4081/23/17
Diagonal 1 function	2	6/0.4015/25/18	7/0.4085/29/21	4/0.4207/18/13
Six Hump function	2	7/0.4112/31/23	7/0.4271/31/23	6/0.4064/27/20
Three Hump function	2	11/0.4441/52/40	11/0.4335/50/38	10/0.4397/47/36
Booth function	2	3/0.3930/13/9	2/0.4043/9/6	2/0.3919/9/6
Zett1 function	2	26/0.5067/105/78	30/0.5581/126/95	24/0.5046/102/77

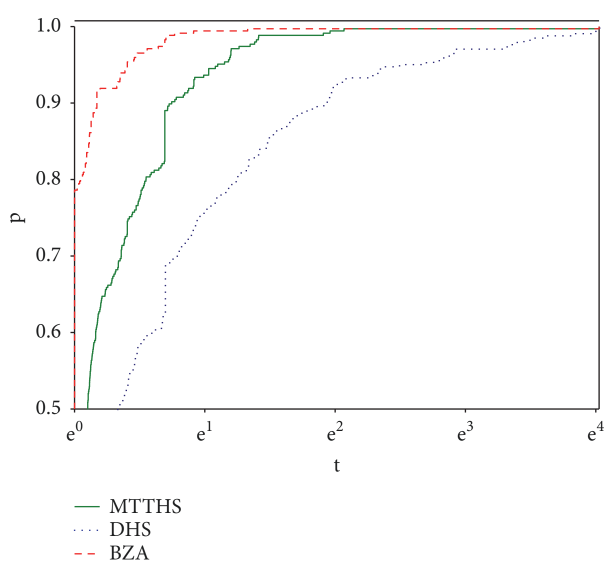

Figure 4: Performance profiles based on function evaluation.

4. Conclusion

We have proposed a modified three-term HS conjugate gradient method. An attractive property of the proposed method is that it produces a sufficient descent condition $g_k^T d_k = -\|g_k\|^2$, regardless of the line search. The global convergence properties of the proposed method have been established under Wolfe line search conditions. Numerical results show that the proposed method is more efficient and robust than state-of-the-art three term (MTTHS) and two-term (DHS) CG methods.

Data Availability

No data were used to support this study.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

References

[1] E. Polak, *Optimization: Algorithms and Consistent Approximations*, Springer, New York, NY, USA, 1997.

[2] J. Nocedal, "Conjugate gradient methods and nonlinear optimization," in *Linear and Nonlinear Conjugate Gradient Related Methods*, L. Adams and J. L. Nazareth, Eds., pp. 9–23, SIAM, Philadelphia, PA, USA, 1995.

[3] R. Ziadi, R. Ellaia, and A. Bencherif-Madani, "Global optimization through a stochastic perturbation of the Polak-Ribière conjugate gradient method," *Journal of Computational and Applied Mathematics*, vol. 317, pp. 672–684, 2017.

[4] M. R. Hestenes and E. Stiefel, "Methods of conjugate gradients for solving linear systems," *Journal of Research of the National Bureau of Standards*, vol. 49, pp. 409–436, 1952.

[5] R. Fletcher and C. M. Reeves, "Function minimization by conjugate gradients," *The Computer Journal*, vol. 7, pp. 149–154, 1964.

[6] E. Polak and G. Ribière, "Note sur la convergence de méthodes de directions conjuguées," *Revue Française d'Informatique et de Recherche Opérationnelle*, vol. 3, no. 16, pp. 35–43, 1969.

[7] B. T. Polyak, "The conjugate gradient method in extreme problems, USSR Comput," *USSR Computational Mathematics and Mathematical Physics*, vol. 9, pp. 94–112, 1969.

[8] R. Fletcher, *Practical Methods of Optimization*, vol. I: Unconstrained Optimization, John Wiley & Sons, New York, NY, USA, 2nd edition, 1987.

[9] Y. Liu and C. Storey, "Efficient generalized conjugate gradient algorithms, Part 1," *Journal of Optimization Theory and Applications*, vol. 69, no. 1, pp. 129–137, 1991.

[10] Y. H. Dai and Y. Yuan, "A nonlinear conjugate gradient method with a strong global convergence property," *SIAM Journal on Optimization*, vol. 10, no. 1, pp. 177–182, 1999.

[11] Z.-J. Shi and J. Guo, "A new family of conjugate gradient methods," *Journal of Computational and Applied Mathematics*, vol. 224, no. 1, pp. 444–457, 2009.

[12] G. Yuan, X. Lu, and Z. Wei, "A conjugate gradient method with descent direction for unconstrained optimization," *Journal of Computational and Applied Mathematics*, vol. 233, no. 2, pp. 519–530, 2009.

[13] Z.-J. Shi, S. Wang, and Z. Xu, "The convergence of conjugate gradient method with nonmonotone line search," *Applied Mathematics and Computation*, vol. 217, no. 5, pp. 1921–1932, 2010.

[14] A. Alhawarat and Z. Salleh, "Modification of nonlinear conjugate gradient method with weak Wolfe-Powell line search," *Abstract and Applied Analysis*, Article ID 7238134, 6 pages, 2017.

[15] Z. Salleh and A. Alhawarat, "An efficient modification of the Hestenes-Stiefel nonlinear conjugate gradient method with restart property," *Journal of Inequalities and Applications*, vol. 2016, no. 1, Article ID 110, 2016.

[16] A. Alhawarat, Z. Salleh, M. Mamat, and M. Rivaie, "An efficient modified Polak-Ribière-Polyak conjugate gradient method with global convergence properties," *Optimization Methods and Software*, vol. 32, no. 6, pp. 1299–1312, 2017.

[17] A. Alhawarat, M. Mamat, M. Rivaie, and Z. Salleh, "An efficient hybrid conjugate gradient method with the strong wolfe-powell line search," *Mathematical Problems in Engineering*, vol. 2015, Article ID 103517, 7 pages, 2015.

[18] E. M. L. Beale, "A derivative of conjugate gradients," in *Numerical Methods for Nonlinear Optimization*, F. A. Lootsma, Ed., pp. 39–43, Academic Press, London, UK, 1972.

[19] L. Nazareth, "A conjugate direction algorithm without line searches," *Journal of Optimization Theory and Applications*, vol. 23, no. 3, pp. 373–387, 1977.

[20] Y. H. Dai and Y. Yuan, *Nonlinear Conjugate Gradient Methods*, Shanghai Science and Technology Publisher, Shanghai, China, 2000.

[21] W. W. Hager and H. Zhang, "A survey of nonlinear conjugate gradient methods," *Pacific Journal of Optimization. An International Journal*, vol. 2, no. 1, pp. 35–58, 2006.

[22] M. F. McGuire and P. Wolfe, "Evaluating a restart procedure for conjugate gradients," Report RC-4382, IBM Research Center, Yorktown Heights, 1973.

[23] N. Y. Deng and Z. Li, "Global convergence of three terms conjugate gradient methods," *Optimization Methods and Software*, vol. 4, pp. 273–282, 1995.

[24] L. Zhang, W. Zhou, and D. H. Li, "A descent modified Polak-Ribiere-Polyak conjugate gradient method and its global convergence," *IMA Journal of Numerical Analysis (IMAJNA)*, vol. 26, no. 4, pp. 629–640, 2006.

[25] L. Zhang, W. Zhou, and D. Li, "Some descent three-term conjugate gradient methods and their global convergence," *Optimization Methods and Software*, vol. 22, no. 4, pp. 697–711, 2007.

[26] W. Cheng, "A two-term PRP-based descent method," *Numerical Functional Analysis and Optimization*, vol. 28, no. 11, pp. 1217–1230, 2007.

[27] A. Y. Al-Bayati and W. H. Sharif, "A new three-term conjugate gradient method for unconstrained optimization," *Canadian Journal on Science and Engineering Mathematics*, vol. 1, no. 5, pp. 108–124, 2010.

[28] J. Zhang, Y. Xiao, and Z. Wei, "Nonlinear conjugate gradient methods with sufficient descent condition for large-scale unconstrained optimization," *Mathematical Problems in Engineering*, vol. 2009, Article ID 243290, 16 pages, 2009.

[29] N. Andrei, "A modified Polak-Ribière-Polyak conjugate gradient algorithm for unconstrained optimization," *Optimization. A Journal of Mathematical Programming and Operations Research*, vol. 60, no. 12, pp. 1457–1471, 2011.

[30] N. Andrei, "On three-term conjugate gradient algorithms for unconstrained optimization," *Applied Mathematics and Computation*, vol. 219, no. 11, pp. 6316–6327, 2013.

[31] N. Andrei, "A simple three-term conjugate gradient algorithm for unconstrained optimization," *Journal of Computational and Applied Mathematics*, vol. 241, pp. 19–29, 2013.

[32] K. Sugiki, Y. Narushima, and H. Yabe, "Globally convergent three-term conjugate gradient methods that use secant conditions and generate descent search directions for unconstrained optimization," *Journal of Optimization Theory and Applications*, vol. 153, no. 3, pp. 733–757, 2012.

[33] Y. Narushima, H. Yabe, and J. A. Ford, "A three-term conjugate gradient method with sufficient descent property for unconstrained optimization," *SIAM Journal on Optimization*, vol. 21, no. 1, pp. 212–230, 2011.

[34] S. Babaie-Kafaki and R. Ghanbari, "Two modified three-term conjugate gradient methods with sufficient descent property," *Optimization Letters*, vol. 8, no. 8, pp. 2285–2297, 2014.

[35] M. Al-Baali, Y. Narushima, and H. Yabe, "A family of three-term conjugate gradient methods with sufficient descent property for unconstrained optimization," *Computational Optimization and Applications*, vol. 60, no. 1, pp. 89–110, 2015.

[36] M. Sun and J. Liu, "Three modified Polak-Ribière-Polyak conjugate gradient methods with sufficient descent property," *Journal of Inequalities and Applications*, vol. 2015, no. 1, 2015.

[37] B. Baluch, Z. Salleh, A. Alhawarat, and U. A. M. Roslan, "A New Modified Three-Term Conjugate Gradient Method with Sufficient Descent Property and Its Global Convergence," *Journal of Mathematics*, Article ID 2715854, 12 pages, 2017.

[38] Z.-F. Dai, "Two modified HS type conjugate gradient methods for unconstrained optimization problems," *Nonlinear Analysis: Theory, Methods & Applications*, vol. 74, no. 3, pp. 927–936, 2011.

[39] Z. Wei, G. Li, and L. Qi, "New nonlinear conjugate gradient formulas for large-scale unconstrained optimization problems," *Applied Mathematics and Computation*, vol. 179, no. 2, pp. 407–430, 2006.

[40] Z. Dai and F. Wen, "Another improved Wei–Yao–Liu nonlinear conjugate gradient method with sufficient descent property," *Applied Mathematics and Computation*, vol. 218, no. 14, pp. 7421–7430, 2012.

[41] J. C. Gilbert and J. Nocedal, "Global convergence properties of conjugate gradient methods for optimization," *SIAM Journal on Optimization*, vol. 2, no. 1, pp. 21–42, 1992.

[42] Y.-H. Dai and L.-Z. Liao, "New conjugacy conditions and related nonlinear conjugate gradient methods," *Applied Mathematics & Optimization*, vol. 43, no. 1, pp. 87–101, 2001.

[43] G. Zoutendijk, "Nonlinear programming, computational methods," in *Integer and Nonlinear Programming*, J. Abadie, Ed., pp. 37–86, North-Holland, Amsterdam, The Netherlands, 1970.

[44] Z.-f. Dai and B.-S. Tian, "Global convergence of some modified PRP nonlinear conjugate gradient methods," *Optimization Letters*, vol. 5, no. 4, pp. 615–630, 2011.

[45] W. W. Hager and H. Zhang, "A new conjugate gradient method with guaranteed descent and an efficient line search," *SIAM Journal on Optimization*, vol. 16, no. 1, pp. 170–192, 2005.

[46] D. C. Liu and J. Nocedal, "On the limited memory BFGS method for large scale optimization," *Mathematical Programming*, vol. 45, no. 1–3, pp. 503–528, 1989.

[47] J. J. Moré, B. S. Garbow, and K. E. Hillstrom, "Testing unconstrained optimization software," *ACM Transactions on Mathematical Software*, vol. 7, no. 1, pp. 17–41, 1981.

[48] N. Andrei, "An unconstrained optimization test functions collection," *Advanced Modeling and Optimization*, vol. 10, no. 1, pp. 147–161, 2008.

[49] E. D. Dolan and J. J. Moré, "Benchmarking optimization software with performance profiles," *Mathematical Programming*, vol. 91, no. 2, pp. 201–213, 2002.