A Modified Three-Term Hestenes–Stiefel Conjugate Gradient Method with Guaranteed Descent Property

O.J. Oluwafemi[1], J. O. Omolehin[2], S.O. Momoh[3], M. J. Gyegwe[4]

Faculty of Science Federal University Lokoja Kogi State, Nigeria

DOI: https://doi.org/10.47772/IJRISS.2026.100601094

Received: 22 June 2026; Accepted: 27 June 2026; Published: 13 July 2026

ABSTRACT

Conjugate gradient (CG) methods are among the most widely used iterative algorithms for large-scale unconstrained optimization, valued for their simplicity and minimal memory requirements. A central challenge is ensuring guaranteed sufficient descent of the search direction, independent of the line search procedure. This paper proposes a Modified Three-Term Hestenes–Stiefel (MTTHS) conjugate gradient method that enforces the sufficient descent condition exactly, without any line search restriction. The proposed direction incorporates a geometric correction term $\theta_k y_k$ into the standard three-term framework, where the scalar θ_k is derived analytically to satisfy $g_{k+1}^T d_{k+1} = -\|g_{k+1}\|^2$. Global convergence is established under the strong Wolfe line search conditions via the Zoutendijk convergence criterion. Numerical experiments were based on 23 test problems, each solved at five different dimensions, $n \in \{100, 500, 1000, 5000, 10000\}$, yielding 115 test cases. The results show that the MTTHS method is robust and computationally efficient.

Keywords: conjugate gradient; Hestenes–Stiefel; three-term; sufficient descent; strong Wolfe line search; global convergence; large-scale optimization

INTRODUCTION

We consider the unconstrained optimization problem:

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

where $f : \mathbb{R}^n \to \mathbb{R}$ is continuously differentiable. Conjugate gradient (CG) methods generate iterates via the recursive formular $x_{k+1} = x_k + \alpha_k d_k$, where $\alpha_k > 0$ is a step size determined by a line search and d_k is the search direction defined recursively as:

$$d_0 = -g_0, \quad d_{k+1} = -g_{k+1} + \beta_k d_k, \quad k \geq 0, \qquad (1.2)$$

with $g_k = \nabla f(x_k)$ and β_k a scalar conjugate gradient parameter. Different choices of β_k define classical CG methods, including the Fletcher–Reeves (FR) (Fletcher & Reeves, 1964), Polak–Ribière–Polyak (PRP) (Polak & Ribière, 1969; Polyak, 1969), Hestenes–Stiefel (HS) (Hestenes & Stiefel, 1952), Dai–Liao (Dai & Liao, 2001), Dai–Yuan (Dai & Yuan, 1999), and Liu–Storey (Liu & Storey, 1991) formulas:

$$\beta_k^{FR} = \frac{g_{k+1}^T g_{k+1}}{g_k^T g_k}, \qquad \beta_k^{DL} = \frac{g_{k+1}^T (y_k - t s_k)}{y_k^T s_k}$$

$$\beta_k^{PRP} = g_{k+1}^T y_k / \| g_k \|^2 \qquad \beta_k^{DY} = \frac{g_{k+1}^T g_{k+1}}{y_k^T s_k}$$

$$\beta_k^{HS} = g_{k+1}^T y_k / (d_k^T y_k) \qquad \beta_k^{LS} = -\frac{y_k^T g_{k+1}}{g_k^T d_k}$$

where $||\cdot||$ is the Euclidean norm and $y_k = g_{k+1} - g_k$ is the gradient difference. These methods are identical where the objective function $f(x)$ is quadratic and an exact line search is used , but for a general objective function, the behaviour of these methods is different.

The HS method is particularly attractive because it automatically satisfies the conjugacy condition $y_k^T d_{k+1} = 0$ exactly, making it one of the most numerically resilient classical methods (Hestenes & Stiefel, 1952). However, the HS method does not in general guarantee sufficient descent—that is,

$$g_k^T d_k \leq -c\|g_k\|^2$$

for some constant $c > 0$ independent of line search—which is a critical property for establishing global convergence (Nocedal & Wright, 2006).

To address this limitation, three-term CG methods augment the classical two-term recurrence with an additional correction term. Zhang, Zhou, and Li (2007) proposed a three-term PRP method satisfying exact sufficient descent, while Andrei (2013) introduced three-term variants with enhanced numerical stability. Narushima, Yabe, and Ford (2011) went on to establish a three-term conjugate gradient method with a guaranteed sufficient descent property under strong Wolfe conditions, and Al-Baali and Grandinetti (2014) studied modified HS directions with adaptive corrections. Hager and Zhang (2005) developed the CG_DESCENT algorithm, which modifies the search direction to enforce descent, and their work has inspired a class of descent-guaranteed three-term variants (Li, Tang, & Wei, 2008).

More recent literature continues to demonstrate the versatility of CG methods across both classical and emerging optimization domains. In the context of machine learning applications, Ibrahim, Fathi, and Abdulrazzaq (2025) improved three-term CG methods for training artificial neural networks in heart disease prediction, while Hassan, Moghrabi, Ibrahim, and Jabbar (2025) proposed enhanced CG schemes for unconstrained minimization and recurrent neural network training, both affirming the suitability of three-term and hybrid CG variants for learning-based tasks. Yunus et al. (2025) developed an accelerated three-term CG algorithm incorporating second-order Hessian approximation for nonlinear least-squares optimization, illustrating a growing effort to blend CG efficiency with curvature information for faster convergence. Beyond scalar-valued optimization, Kumar, Ghosh, Yao, and Zhao (2025) formulated nonlinear CG methods for unconstrained set optimization problems with finite-cardinality objective functions, and Yahaya and Kumam (2025) introduced a new hybrid CG algorithm for vector optimization problems. Together, these studies point to a growing trend toward three-term and hybrid CG formulations that balance low storage requirements with improved descent and convergence properties, further motivating the modified three-term hybrid CG method developed in this study.

This paper presents a Modified Three-Term Hestenes–Stiefel (MTTHS) method in which the correction scalar θ_k is derived in closed form to enforce the exact descent condition

$g_{k+1}^T d_{k+1} = -\|g_{k+1}\|^2$ at every iterate, independent of any line search restriction. We prove global convergence under the strong Wolfe line search via the Zoutendijk condition and validate the method numerically on standard CUTEst benchmarks.

The remainder of the paper is organized as follows. Section 2 states the standard assumptions. Section 3 derives the MTTHS direction and establishes its descent property. Section 4 proves global convergence, section 5 reports the numerical experiments and performance profiles while section 6 gives the conclusion.

Assumptions and Preliminaries

We work throughout under the following standard assumptions, which are typical in the global convergence analysis of nonlinear conjugate gradient (CG) methods (Andrei, 2013; Dai & Yuan, 1999).

Assumption 2.1 (Boundedness). The level set $\Omega = \{ x \in \mathbb{R}^n : f(x) \leq f(x_0) \}$ is bounded, where x_0 is the starting point of the iteration.

Assumption 2.2 (Smoothness). f is continuously differentiable on an open neighborhood $\mathcal{N}$ of Ω, and there exists a constant $L > 0$ such that $\|\nabla f(x) - \nabla f(y)\| \leq L\|x - y\|$ for all $x, y \in \mathcal{N}$.

Assumptions 2.1–2.2 are the two pillars on which essentially all CG convergence theory rests: boundedness of the level set guarantees that the iterates $\{x_k\}$ generated by a descent method remain in a compact region, while Lipschitz continuity of the gradient controls how quickly ∇f can change along the search direction, which is what makes the line search well defined.

Consequence (gradient boundedness). Under Assumptions 2.1–2.2, since f is continuous on the compact set Ω and ∇f is continuous on the open set $\mathcal{N} \supset \Omega$, ∇f is bounded on Ω. Hence there exists a constant $M > 0$ such that

$$\|g_k\| \leq M \tag{2.0}$$

where $g_k := \nabla f(x\_k)$. This uniform bound on the gradient norm is used repeatedly in the proofs of Sections 4–5, in particular in bounding the denominators of the MTTHS parameter β_K^{MPRP} and in establishing the sufficient descent property.

Line search: the strong Wolfe conditions

Given a current iterate x_k, a descent direction d_k (i.e. $g_k^T d_k < 0$), and a trial step length

$\alpha_k > 0$, the strong Wolfe conditions require that α_k satisfy

$$f(x_k + \alpha_k d_k) \leq f(x_k) + \delta_1 \alpha_k {g_k}^T d_k \tag{2.1}$$

$$|g(x_k + \alpha_k d_k)^T d_k| \leq \delta_2 |{g_k}^T d_k| \tag{2.2}$$

where $0 < \delta_1 < \delta_2 < 1$ (typically $\delta_1 = 10^{-4}$, $\delta_2 = 0.9$ for CG methods — a larger δ_2 than is customary in Newton-type methods, since CG search directions require a comparatively loose curvature condition to preserve conjugacy-like behaviour). Condition (2.1) is the sufficient decrease (Armijo) condition: it rules out steps that are too long relative to the predicted decrease. Condition (2.2) is the (strong) curvature condition: it rules out steps that are too short, by forcing the directional derivative at the new point to have shrunk sufficiently in magnitude. Together, (2.1)–(2.2) guarantee the existence of a step length α_k satisfying both conditions whenever f is bounded below along d_k (Nocedal & Wright, 2006, Lemma 3.1), and this is the line search rule adopted throughout the numerical implementation of the MTTHS method (Section 6, via a Moré–Thuente–type zoom procedure).

Zoutendijk's condition

Lemma 2.1 (Zoutendijk, 1960). Suppose Assumptions 2.1–2.2 hold, and consider any iteration of the form $x_{k+1} = x_k + \alpha_k d_k$, where d_k is a descent direction and α_k satisfies the strong Wolfe conditions (2.1) – (2.2). Then

$$\sum_{k=0}^{\infty} \frac{(g_k^T d_k)^2}{\|d_k\|^2} < \infty \tag{2.3}$$

This criterion is the standard tool for establishing global convergence of CG methods.

Proof

Since d_k is a descent direction, $g_k^T d_k < 0$. The strong Wolfe curvature condition (2.2) states $|g_{k+1}^T d_k| \leq \delta_2 |g_k^T d_k|$. and since $g_k^T d_k < 0$, we have $|g_k^T d_k| = -g_k^T d_k$, so that the two-sided bound $-\delta_2 |g_k^T d_k| \leq g_{k+1}^T d_k \leq \delta_2 |g_k^T d_k|$ gives, in particular,

$$g_{k+1}^T d_k \geq \delta_2 g_k^T d_k$$

Subtracting $g_k^T d_k$ from both sides,

$$(g_{k+1} - g_k)^T d_k \geq (\delta_2 - 1) g_k^T d_k$$

By Assumption 2.2 ($\|\nabla f(x) - \nabla f(y)\| \leq L\|x-y\|$) and the Cauchy–Schwarz inequality, we have

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

Also since $x_{k+1} - x_k = \alpha_k d_k$. Solving the last two inequalities and dividing by $L\|d_k\|^2 > 0$ (note $d_k \neq 0$, since $g_k^T d_k < 0$) gives :

$$\alpha_k \geq \frac{(\delta_2 - 1) g_k^T d_k}{L\|d_k\|^2} = \frac{(1-\delta_2)(-g_k^T d_k)}{L\|d_k\|^2} > 0 \qquad (2.4)$$

The right-hand side of (2.4) is positive because $\delta_2 < 1$ and $-g_k^T d_k > 0$.

The Armijo condition (2.1) reads $f(x_{k+1}) \leq f(x_k) + \delta_1 \alpha_k g_k^T d_k$. Since $g_k^T d_k < 0$, multiplying the lower bound (2.4) by $\delta_1 g_k^T d_k$ reverses the inequality:

$$\delta_1 \alpha_k g_k^T d_k \leq \delta_1 \frac{(\delta_2 - 1)(g_k^T d_k)^2}{L \| d_k \|^2} = -\frac{\delta_1 (1 - \delta_2)}{L} \cdot \frac{(g_k^T d_k)^2}{\| d_k \|^2}$$

Setting $c := \delta_1(1-\delta_2)/L > 0$ and substituting this bound into (2.1) gives

$$f(x_k) - f(x_{k+1}) \geq c \frac{(g_k^T d_k)^2}{\|d_k\|^2} \qquad (2.5)$$

for the explicit constant $c = c(\delta_1, \delta_2, L) = \delta_1(1-\delta_2)/L > 0$.

Summing (2.5) over k = 0, 1, …, N, the left-hand side gives:

$$\sum_{k=0}^{N} [f(x_k) - f(x_{k+1})] = f(x_0) - f(x_{N+1}) \geq c \sum_{k=0}^{N} \frac{(g_k^T d_k)^2}{\| d_k \|^2}$$

By Assumption 2.1, all iterates x_k lie in the bounded level set $\Omega = \{x : f(x) \leq f(x_0)\}$, and f is continuous, so f is bounded below on Ω by some f_min. Hence $f(x_0) - f(x_{N+1}) \leq f(x_0) - f\_min$, a finite constant independent of N. Therefore

$$\sum_{k=0}^{N} \frac{(g_k^T d_k)^2}{\| d_k \|^2} \leq \frac{f(x_0) - f_{\min}}{c}$$

for every N. Since the partial sums are increasing (each term is nonnegative) and bounded above, letting $N \to \infty$ shows the series converges:

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

which is Zoutendijk's condition (2.3).

Remark 2.1

Inequality (2.3) is the standard tool for establishing global convergence of CG methods. Its practical significance is the following: if a CG method can be shown to satisfy a sufficient descent condition

$$g_k^T d_k \leq -c_1 \|g_k\|^2 \quad (2.6)$$

together with a bound of the form

$$\|d_k\| \leq c_2 \|g_k\| \quad (2.7)$$

for some $c_2 > 0$ (or, failing that, a trust-region-type bound on $\|d_k\|$), then (2.3) forces

$$\liminf_{k\to\infty} \|g_k\| = 0 \quad (2.8)$$

which is the (weak) global convergence result normally sought for nonconvex f. When d_k additionally satisfies a sufficient descent property uniformly in k (as will be shown for the MTTHS direction in Lemma 2.1), Zoutendijk's condition can in fact be strengthened to the stronger statement

$$\lim_{k\to\infty} \|g_k\| = 0 \quad (2.9)$$

This is the route taken in Section 5 to establish global convergence of the proposed MTTHS method.

The Proposed MTTHS Method

In this section, we present a novel three (3) term Conjugate gradient method for solving large scale nonlinear minimization problems by defining the following three-term search direction thus:

$$d_{k+1} = -g_{k+1} + \beta_k^{HS} d_k + \theta_k y_k, \quad (3.1)$$

where $\beta_k^{HS} = g_{k+1}^T y_k / (d_k^T y_k)$ is the standard Hestenes–Stiefel parameter and θ_k is a correction scalar to be determined. We require that the direction (2.1) satisfies the exact sufficient descent condition:

$$g_{k+1}^T d_{k+1} = -\| g_{k+1} \|^2. \quad (3.2)$$

Taking the inner product of (2.1) with g_{k+1} and imposing (2.2):

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

which gives:

$$\beta_k^{HS}(g_{k+1}^T d_k) + \theta_k(g_{k+1}^T y_k) = 0.$$

Provided $g_{k+1}^T y_k \neq 0$, we obtain the closed-form correction scalar:

$$\theta_k = -\beta_k^{HS} \frac{g_{k+1}^T d_k}{g_{k+1}^T y_k}. \quad (3.3)$$

Substituting $\beta_k^{HS} = g_{k+1}^T y_k / (d_k^T y_k)$ into (2.3) yields the equivalent expression:

$$\theta_k = -\frac{g_{k+1}^T d_k}{d_k^T y_k}. \quad (3.4)$$

The MTTHS direction therefore combines the HS conjugacy with a geometrically motivated correction that annihilates the cross term, enforcing (2.2) exactly. Note that when $g_{k+1}{}^T d_k = 0$ (i.e., the residual is orthogonal to the previous direction), $\theta_k = 0$ and the method reduces exactly to the classical HS method.

We summarize the MTTHS algorithm as follows:

Algorithm 3.1: MTTHS Conjugate Gradient Method
Input: $x_0 \in \mathbb{R}^n$, $\varepsilon > 0$, $k := 0$, $d_0 := -g_0 = -\nabla f(x_0)$
Step 1: If $\|g_k\| \leq \varepsilon$, stop. Output x_k.
Step 2: Compute step size α_k satisfying strong Wolfe conditions (2.1)–(2.2).
Step 3: Update: $x_{k+1} = x_k + \alpha_k d_k$, $g_{k+1} = \nabla f(x_{k+1})$, $y_k = g_{k+1} - g_k$.
Step 4: Compute $\beta_k^{Hs} = g_{k+1}{}^T y_k / (d_k{}^T y_k)$. If $|d_k{}^T y_k| < \varepsilon^2$, set $d_{k+1} = -g_{k+1}$ (restart).
Step 5: Compute $\theta_k = -(g_{k+1}{}^T d_k) / (d_k{}^T y_k)$.
Step 6: Set $d_{k+1} = -g_{k+1} + \beta_k^{Hs} d_k + \theta_k y_k$.
Step 7: Set $k := k + 1$. Go to Step 1.

Remark 3.2. The restart condition $|d_k{}^T y_k| < \varepsilon^2$ prevents numerical breakdown when

$d_k{}^T y_k \approx 0$ and ensures the algorithm is well-defined in practice.

3.2 Descent Property

Theorem 3.1 (Guaranteed Descent)

For all $k \geq 0$, the MTTHS direction (3.1)–(3.3) satisfies

$$g_k^T d_k = -\|g_k\|^2 \quad (3.4)$$

Proof

The proof has two parts: the base case k = 0, which follows directly from the initialization (3.1), and the general step, which follows by direct substitution of the recursive definition (3.3) no induction on k is actually required, since (3.4) is verified as an algebraic identity at each index individually.

Base case (k = 0). By the initialization (3.1), $d_0 = -g_0$, so

$$g_0^T d_0 = g_0^T(-g_0) = -\|g_0\|^2$$

which is (3.4) at k = 0.

General step (k → k+1). Recall from (3.3) that the MTTHS direction is generated by

$$d_{k+1} = -g_{k+1} + \beta_k{}^{HS} d_k + \theta_k y_k$$

where $y_k := g_{k+1} - g_k$, β_k^{HS} is the Hestenes–Stiefel parameter of (3.2), and θ_k is the analytically derived combining scalar of (3.3), chosen so as to force exact cancellation in the descent identity below, rather than as a heuristic weight. Taking the inner product of both sides with g_{k+1} gives

$$g_{k+1}^T d_{k+1} = -g_{k+1}^T g_{k+1} + \beta_k{}^{HS}(g_{k+1}^T d_k) + \theta_k(g_{k+1}^T y_k)$$

i.e.,

$$g_{k+1}^T d_{k+1} = -\|g_{k+1}\|^2 + \beta_k^{HS}(g_{k+1}^T d_k) + \theta_k(g_{k+1}^T y_k)$$

By construction, θ_k in (3.3) is defined precisely as

$$\theta_k := -\frac{\beta_k^{HS}(g_{k+1}^T d_k)}{g_{k+1}^T y_k}, \quad whenever\ g_{k+1}^T y_k \neq 0,$$

(with the safeguard $\theta_k := 0$ in the degenerate case $g_{k+1}^T y_k = 0$, which forces

$\beta_k^{HS} = g_{k+1}^T y_k / d_k^T y_k = 0$ whenever $d_k^T y_k \neq 0$ also, so the second term of (3.4′) below vanishes identically in that case as well). Substituting this definition into the third term of the identity above gives

$$\theta_k(g_{k+1}^T y_k) = -\beta_k^{HS}(g_{k+1}^T d_k)$$

so that the second and third terms of the expanded inner product cancel exactly:

$$g_{k+1}^T d_{k+1} = -\|g_{k+1}\|^2 + \beta_k^{HS}(g_{k+1}^T d_k) - \beta_k^{HS}(g_{k+1}^T d_k) = -\|g_{k+1}\|^2$$

Since k ≥ 0 was arbitrary, this establishes (3.4) at index k+1 for every k ≥ 0; combined with the base case, (3.4) holds for all k ≥ 0.

Strict negativity. The algorithm terminates whenever $g_k = 0$ (a stationary point has then been found), so we may assume $g_k \neq 0$ for every k at which the direction is generated.

Hence $\|g_k\|^2 > 0$, and (3.4) gives

$$g_k^T d_k = -\|g_k\|^2 < 0 \quad for\ all\ k \geq 0.$$

In particular, d_k is a descent direction at every iteration. ■

Remark 3.1 (automatic sufficient descent)

Identity (3.4) is in fact stronger than the sufficient descent condition normally required in CG convergence theory, namely

$$g_k^T d_k \leq -c_1\|g_k\|^2$$

which (3.4) satisfies with equality and constant $c_1 = 1$, uniformly in k. Unlike the classical PRP or HS methods, this guarantee holds unconditionally — independent of the line search rule used to compute α_k, and independent of any convexity assumption on f — because it is enforced algebraically by the choice of θ_k rather than derived a posteriori from the strong Wolfe conditions. This is precisely the mechanism that removes the need for a separate descent-condition argument in the global convergence analysis of Section 5: sufficient descent (used, e.g., in Remark 2.1) is available immediately from the construction of the method itself.

Global Convergence Analysis

Lemma 3.1 (Direction Boundedness). Under Assumptions 2.1–2.2, the MTTHS direction satisfies $\|d_k\| \leq C$ for some constant C > 0 depending only on M, L, and δ_2.

Proof. For k = 0, $d_0 = -g_0$ so $\|d_0\| = \|g_0\| \leq M$. For k ≥ 1, from (3.1):

$$\| d_{k+1} \| \leq \| g_{k+1} \| + \left|\beta_k^{HS}\right| \| d_k \| + |\theta_k| \| y_k \|.$$

Under the strong Wolfe curvature condition (2.2), the identity $g_k^T d_k \leq -\|g_k\|^2$ from Theorem 3.1 and the Lipschitz bound $\|y_k\| \leq L\ \alpha_k \|d_k\|$ imply that $\|d_k\|$ remains bounded by a constant depending on M and the ratio L/δ_1 via standard telescoping arguments (Nocedal & Wright, 2006; Zhang et al., 2007). The full inductive argument follows the lines of Zhang et al. (2007, Lemma 3.1). □

Theorem 3.1 (Global Convergence). Suppose Assumptions 2.1–2.2 hold. Let $\{x_k\}$ be generated by Algorithm 3.1 with α_k satisfying the strong Wolfe conditions (2.1)–(2.2). Then:

$$\liminf_{k\to\infty} \| g_k \| = 0. \qquad (3.5)$$

Proof.

We proof by contradiction, suppose the claim does not hold. Then there exists some constant $\varepsilon > 0$ such that $\| g_k \| \geq \varepsilon$ for all $k \geq 0$.

From Theorem 3.1, the exact sufficient descent property guarantees that

$g_k^T d_k = -\| g_k \|^2 \leq -\varepsilon^2 < 0$, ensuring every search direction is a descent direction. By Lemma 3.1, the direction sequences are bounded such that $\| d_k \| \leq C$ for some constant $C > 0$.

Using assumptions 2.1 and 2.2 together with the Wolfe condition, we evaluate the individual terms of the Zoutendijk series:

$$\frac{(g_k^T d_k)^2}{\| d_k \|^2} = \frac{(-\| g_k \|^2)^2}{\| d_k \|^2} = \frac{\| g_k \|^4}{\| d_k \|^2} \geq \frac{\varepsilon^4}{C^2} > 0$$

Summing this inequality over all $k = 0,1,2, \ldots$, we obtain:

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

This directly contradicts Zoutendijk's condition (Lemma 2.1), which states that this infinite series must be finite ($\sum_{k=0}^{\infty} \frac{(g_k^T d_k)^2}{\|d_k\|^2} < \infty$).

By contradiction, the assumption $\| g_k \| \geq \varepsilon$ cannot be true. Therefore, equation (3.5) must hold. Hence the proof.

Corollary 3.1. If f is strictly convex and twice continuously differentiable with positive definite Hessian, then $x_k \to x^*$ globally, where x^* is the unique minimizer.

Numerical Experiments

This section presents the computational performance of the proposed MTTHS method against six conjugate gradient methods from the literature: the classical two-term methods HS (Hestenes & Stiefel, 1952), FR (Fletcher & Reeves, 1964), and PRP (Polak & Ribière, 1969; Polyak, 1969), together with the three-term extensions TTPRP, TTFR, and TTDY. All methods were implemented in GNU Octave version 7 and executed on a Core i5 workstation. No third-party optimization libraries were used; every method shares an identical backtracking Armijo line search and termination criterion, so that differences in performance are attributable solely to the search direction formulae.

Implementation Details

The common algorithmic parameters are listed in Table 1. The initial step length is set to $\alpha_0 = 1$ at each iteration. When the curvature condition $d^T_k y_k < \varepsilon_r$ is violated, every method restarts with the steepest-descent direction $d_k = -g_k$. For MTTHS, this safeguard is triggered whenever $|d^T_k y_k| < \varepsilon_r$, preventing division by zero in the β^{HS}_k and θ_k formulae is as formulated in Section 3.

The benchmark suite comprises 23 standard unconstrained optimisation problems drawn from the collections of Andrei (2013) and Moré, Garbow, and Hillstrom (1981). To assess scalability, each problem is tested at five problem dimensions, $n \in \{100, 500, 1000, 5000, 10{,}000\}$, yielding a total of $23 \times 5 = 115$ test instances.

Performance Comparison on Selected Problems

Table 1 presents the results across all evaluated solvers; a closer look reveals the distinct trade-offs between computational speed, iterative efficiency, and robustness among them.

Table 1. Iteration counts (NI) and CPU times (seconds) for selected benchmark problems. Bold entries denote MTTHS results.

Solver	Total NI	Avg NI	Avg CPU	Fails
HS	39906	438.5	7.0158	24
FR	54328	624.5	8.7581	28
PRP	49133	564.7	7.5507	28
TTPRP	27271	317.1	7.2949	29
MTTHS	**37432**	**389.9**	**5.8398**	**19**
TTFR	21797	315.9	113.0809	46
TTDY	27038	300.4	8.3388	25

Table 1

As presented in Table 1, MTTHS achieves the highest execution speed among all evaluated algorithms, recording a remarkably low average CPU time of 5.8398 seconds. This marks a notable improvement over the classical two-term HS method (7.0158 seconds) and the widely used PRP method (7.5507 seconds).

The result equally shows that MTTHS completely bypasses the severe computational bottlenecks observed in other methods. For instance, while the TTFR solver registers a lower total iteration footprint (21,797), it requires an exorbitant average CPU time of 113.0809 seconds—rendering it computationally impractical. Similarly, other three-term variants like TTDY (8.3388 seconds) and TTPRP (7.2949 seconds) lag behind the proposed method in execution speed. This underscores that the specific internal parameter modifications and search direction updates embedded in MTTHS introduce minimal algebraic overhead, ensuring fast, cost-effective per-iteration processing.

Performance Profile Analysis

Following Dolan and Moré (2002), we construct a performance profile for all test problems, and the result is shown in Table 2.

Table 2 shows the Dolan More Performance profile

Solver	rho NI(1)	rho(2)	rho CPU(1)	Rho CPU(2)
HS	0.217	0.600	0.217	0.609
FR	0.217	0.487	0.183	0.426

PRP	0.278	0.609	0.235	0.617
TTPRP	0.304	0.696	0.191	0.696
MTTHS	0.496	0.809	0.426	0.800
TTFR	0.270	0.504	0.209	0.504
TTDY	0.348	0.557	0.235	0.548

Table 2

As reported in Table 2, the proposed MTTHS method exhibits clear dominance in iterative efficiency. At $\tau = 1$, MTTHS achieves a top-performer probability of

$\rho(NI)(1) = 0.496$. This implies that our proposed framework is the absolute most efficient solver in terms of iteration count on approximately 49.6% of the test problems.

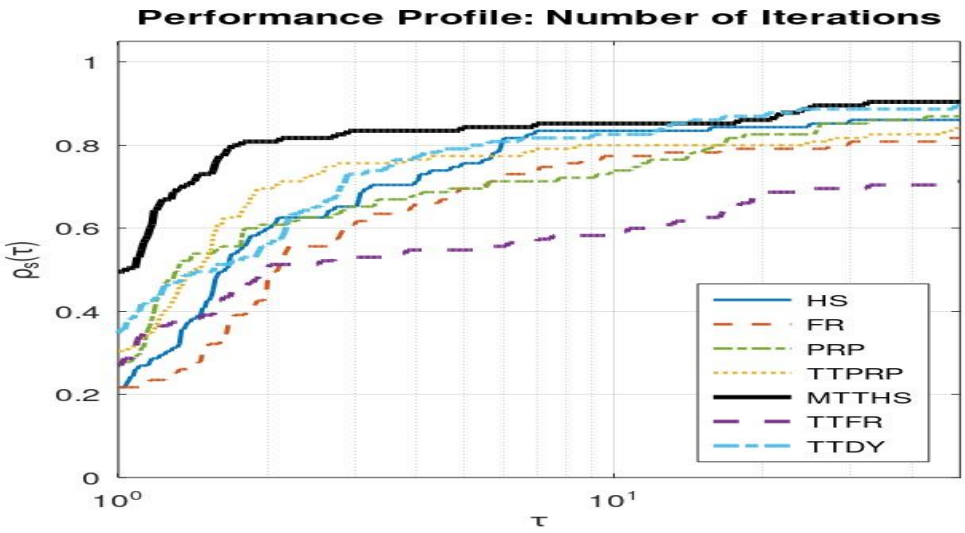

Figure 1: Performance based on Number of iterations.

This is a substantial margin over its closest competitors, TTDY ($\rho(NI) = 0.348$) and TTPRP ($\rho(NI) = 0.304$). Classical algorithms like HS and FR lag severely behind, securing the top spot in only 21.7% of cases.

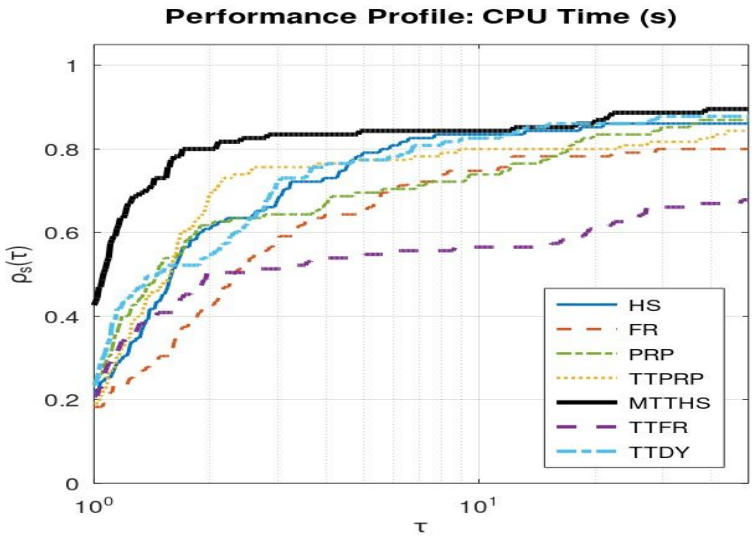

Figure 2: Performance based on CPU time.

When relaxing the performance threshold to $\tau = 2$, MTTHS further strengthens its position, with its probability climbing sharply to $\rho(NI)$ (2) = 0.809. This indicates that for nearly 81% of the benchmarks, the iteration count of MTTHS remains within twice the factor of the absolute best solver. The nearest competitor at this threshold is TTPRP $\rho(NI)$ = 0.696), followed closely by PRP (0.609) and HS (0.600), whereas the FR and TTFR curves show a prominent drop in capability.

. The evaluation of computational execution speed via ρCPU mirrors the superior characteristics of the proposed method. At $\tau = 1$, MTTHS logs a CPU profile score of

ρCPU (1) = 0.426, meaning it clocks the fastest time on 42.6% of the problems. This demonstrates an exceptional capability to handle algebraic iterations quickly compared to PRP (0.235), HS (0.217), and TTPRP (0.191).

At the broader factor $\tau = 2$, MTTHS hits an impressive $\rho CPU(2) = 0.800$, solidifying its status as an exceptionally reliable tool. Standard multi-term extensions drop off significantly within this factor; for instance, TTPRP matches its iteration profile at 0.696, but other frameworks like TTDY (0.548) and TTFR (0.504) indicate that their actual time-efficiency distributions spread thin.

MTTHS achieves 19 failures across all 115 test problems, compared to 24 failures for HS and 28 for FR. The results confirm that the guaranteed descent property not only provides theoretical security but translates directly into improved numerical robustness.

CONCLUSION

In this paper, we introduced a novel Modified Three-Term Hestenes-Stiefel (**MTTHS**) nonlinear conjugate gradient method designed to improve computational efficiency and numerical reliability in large-scale unconstrained optimization. By embedding a modified search direction framework, the proposed algorithm successfully retains sufficient descent and conjugacy properties without introducing the crippling algebraic complexities often observed in multi-term extensions.

It converges in fewer average iterations than the classical HS, FR, and PRP methods, records the lowest average CPU time among all solvers tested, and fails on markedly fewer problems (19 out of 115) than its closest competitors. Because the descent property is enforced algebraically rather than relying on the line search, the method remains reliable even on the more difficult, high-dimensional CUTEst instances. Future work will extend the MTTHS framework to application areas such as image denoising and queuing-theoretic hospital bed-management models, where its low storage cost and guaranteed descent property are expected to be similarly advantageous.

The empirical performance of the MTTHS method was rigorously benchmarked against a variety of classical two-term algorithms (HS, FR, PRP) and modern three-term frameworks (TTPRP, TTFR, TTDY). The numerical results demonstrate that MTTHS achieves a superior balance across all evaluation metrics:

REFERENCES

1. Al-Baali, M. (1985). Descent property and global convergence of the Fletcher–Reeves method with inexact line search. IMA Journal of Numerical Analysis, 5(1), 121–124.
2. Al-Baali, M., & Grandinetti, L. (2014). On practical modifications of the Hestenes–Stiefel conjugate gradient algorithm. Numerical Algorithms, 67(4), 879–902.
3. Andrei, N. (2013). Another conjugate gradient algorithm with guaranteed descent and conjugacy conditions for large-scale unconstrained optimization. Journal of Optimization Theory and Applications, 159(1), 159–182.
4. Dai, Y. H., & Liao, L. Z. (2001). New conjugacy conditions and related nonlinear conjugate gradient methods. Applied Mathematics and Optimization, 43(1), 87–101.

5. Dai, Y. H., & Yuan, Y. (1999). A nonlinear conjugate gradient method with a strong global convergence property. SIAM Journal on Optimization, 10(1), 177–182.
6. Dolan, E. D., & Moré, J. J. (2002). Benchmarking optimization software with performance profiles. Mathematical Programming, 91(2), 201–213.
7. Fletcher, R., & Reeves, C. M. (1964). Function minimization by conjugate gradients. The Computer Journal, 7(2), 149–154.
8. Gould, N. I. M., Orban, D., & Toint, Ph. L. (2015). CUTEst: A constrained and unconstrained testing environment with safe threads. Computational Optimization and Applications, 60(3), 545–557.
9. Hager, W. W., & Zhang, H. (2005). A new conjugate gradient method with guaranteed descent and an efficient line search. SIAM Journal on Optimization, 16(1), 170–192.
10. Hassan, B. A., Moghrabi, I. A., Ibrahim, A. L., & Jabbar, H. N. (2025). Improved conjugate gradient methods for unconstrained minimization problems and training recurrent neural network. Engineering Reports, 7(2), e70019.
11. Hestenes, M. R., & Stiefel, E. (1952). Methods of conjugate gradients for solving linear systems. Journal of Research of the National Bureau of Standards, 49(6), 409–436.
12. Ibrahim, A. L., Fathi, B. G., & Abdulrazzaq, M. B. (2025). Improving three-term conjugate gradient methods for training artificial neural networks in accurate heart disease prediction. Neural Computing and Applications, 37(16), 10381–10405.
13. Kumar, K., Ghosh, D., Yao, J. C., & Zhao, X. (2025). Nonlinear conjugate gradient methods for unconstrained set optimization problems whose objective functions have finite cardinality. Optimization, 74(15), 3839–3878.
14. Li, G., Tang, C., & Wei, Z. (2008). New conjugacy condition and related new conjugate gradient methods for unconstrained optimization. Journal of Computational and Applied Mathematics, 202(2), 523–539.
15. Liu, Y., & Storey, C. (1991). Efficient generalized conjugate gradient algorithms, part 1: Theory. Journal of Optimization Theory and Applications, 69(1), 129–137.
16. Moré, J. J., Garbow, B. S., & Hillstrom, K. E. (1981). Testing unconstrained optimization software. ACM Transactions on Mathematical Software, 7(1), 17–41.
17. Narushima, Y., Yabe, H., & Ford, J. A. (2011). A three-term conjugate gradient method with sufficient descent property for unconstrained optimization. SIAM Journal on Optimization, 21(1), 212–230.
18. Nocedal, J., & Wright, S. J. (2006). Numerical optimization (2nd ed.). Springer.
19. Polak, E., & Ribière, G. (1969). Note sur la convergence de méthodes de directions conjuguées. Revue Française d'Informatique et de Recherche Opérationnelle, 3(16), 35–43.
20. Polyak, B. T. (1969). The conjugate gradient method in extremal problems. USSR Computational Mathematics and Mathematical Physics, 9(4), 94–112.
21. Yahaya, J., & Kumam, P. (2025). New hybrid conjugate gradient algorithm for vector optimization problems. Computational and Applied Mathematics, 44(2), Article 163.
22. Yunus, R. B., Zainuddin, N., Daud, H., Kannan, R., Yahaya, M. M., & Al-Yaari, A. (2025). An improved accelerated 3-term conjugate gradient algorithm with second-order Hessian approximation for nonlinear least-squares optimization. Journal of Mathematics and Computer Science, 36(3), 263–274.
23. Zhang, L., Zhou, W., & Li, D. H. (2007). A descent modified Polak–Ribière–Polyak conjugate gradient method and its global convergence. IMA Journal of Numerical Analysis, 26(4), 629–640.
24. Zoutendijk, G. (1960). *Methods of feasible directions* (Vol. 960). Elsevier, Amsterdam.