

An enhanced BFGS framework with nonmonotone line search for nonconvex unconstrained optimization

Parisa Ostovari, Keyvan Amini*

Department of Mathematics, Faculty of Sciences, Razi University, Kermanshah, Iran

Email(s): parisaostovari71@gmail.com, kamini@razi.ac.ir

Abstract. Quasi-Newton methods constitute an important class of iterative techniques for solving unconstrained optimization problems, with the BFGS method being one of the most successful and widely used members of this class. Despite its well-established effectiveness in solving nonlinear unconstrained optimization problems, the BFGS algorithm still suffers from several limitations that may affect its computational efficiency and performance, particularly for large-scale problems. First, the direction generated by the BFGS method may fail to be a descent direction, particularly when the condition $s_k^T y_k > 0$ fails to hold. Consequently, the positive definiteness of the matrix B_k cannot be guaranteed. Second, although global and superlinear convergence results have been established for convex optimization problems, these results do not necessarily hold for general problems. In this paper, we propose a new modified BFGS method with nonmonotone line search for solving nonconvex unconstrained optimization problems, where the approximate Hessian matrix is updated based on a modified secant condition and an adaptive scaling parameter. It is shown that the search direction is a descent direction independently of the line search. The global convergence of the proposed method has been established under certain standard assumptions, without assuming convexity of the objective function. We also report some numerical experiments to demonstrate the efficiency of the proposed method.

Keywords: Unconstrained optimization, quasi-Newton method, modified BFGS method, nonmonotone line search, global convergence.

AMS Subject Classification 2020: 90C53, 90C30.

1 Introduction

Consider the following unconstrained optimization problem:

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

*Corresponding author

Received: 26 July 2026/ Revised: 16 October 2026/ Accepted: 18 October 2026

DOI: [10.22124/jmm.2026.34273.3179](https://doi.org/10.22124/jmm.2026.34273.3179)

where $f: \mathbb{R}^n \rightarrow \mathbb{R}$ is a continuously differentiable function. Several methods have been proposed for solving (1), including conjugate-gradient, Newton, quasi-Newton, steepest-descent, and trust-region methods; see [25]. Among these, Quasi-Newton methods are widely used because of their convergence properties and favorable numerical behavior. This family of methods generates an iterative sequence $\{x_k\}$ to solve problem (1) in the following form

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

in which $\alpha_k > 0$ is a steplength generated by a suitable line search and d_k is a solution of the system of linear equations

$$B_k d_k = -g_k, \quad (2)$$

where $g_k := \nabla f(x_k)$ is the gradient vector of the function f at x_k , $f_k := f(x_k)$, B_k is an approximation of the Hessian matrix, $G_k = \nabla^2 f_k$ and is generated in such a way that it satisfies the so-called secant conditions

$$B_{k+1} s_k = y_k,$$

while $s_k = x_{k+1} - x_k$ and $y_k = g_{k+1} - g_k$.

Multiple methods have been proposed for generating B_k while the main idea in most approaches is to generate B_{k+1} using B_k and the current problem information. Among these, the *BFGS* method is one of the most effective and widely used methods for generating B_{k+1} [10, 25]. Recent studies have also investigated various modifications of the BFGS method to improve its efficiency, robustness, and convergence properties [8, 11, 14, 20]. The BFGS method uses the following update formula

$$B_{k+1} = B_k - \frac{B_k s_k s_k^T B_k}{s_k^T B_k s_k} + \frac{y_k y_k^T}{s_k^T y_k}. \quad (3)$$

It is known that if B_k be positive definite and the condition $s_k^T y_k > 0$ holds, then B_{k+1} is also positive definite. Therefore, the quasi-Newton direction generated by (2) is a descent direction for the function f at x_k . Various line-search procedures guarantee the condition $s_k^T y_k > 0$. For example, if the steplength α_k satisfies the Wolfe conditions [25]

$$\begin{cases} f(x_k + \alpha_k d_k) \leq f_k + \sigma_0 \alpha_k g_k^T d_k, \\ g(x_k + \alpha_k d_k)^T d_k \geq \sigma_1 g_k^T d_k, \end{cases} \quad (4)$$

in such a way that $0 < \sigma_0 < \sigma_1 < 1$, then it follows that $s_k^T y_k > 0$.

Among the commonly used line-search strategies, the Armijo rule employs the following inequality to compute the steplength α_k

$$f(x_k + \alpha_k d_k) \leq f_k + \sigma \alpha_k g_k^T d_k, \quad (5)$$

where $0 < \sigma < 1$.

The Armijo line search does not guarantee the condition $s_k^T y_k > 0$; therefore B_{k+1} , even with the positive definiteness of B_k , is not necessarily positive definite when Armijo line search is used.

Researchers have demonstrated the convergence of the *BFGS* method for solving convex problems [4, 5, 26]. However, Dai [6] provided an example showing that the standard *BFGS* method with Wolfe line search may fail for nonconvex functions. On the other hand, Mascarenhas [21] showed with an example

that even with exact line search, the *BFGS* method may not converge. To improve the convergence properties of the *BFGS* method, Li and Fukushima [16, 17] proposed a modified *BFGS* algorithm by making changes to the standard *BFGS* update, and under suitable conditions, they showed that this method has global convergence and superlinear convergence for nonconvex optimization problems. The results presented by Li and Fukushima have led many researchers to seek improvements to the *BFGS* method. In 2009, Xiao et al. presented a modified version of the Li and Fukushima algorithm that uses a nonmonotone line search. They showed that this method converges for nonconvex optimization problems [29].

In general, iterative methods are designed so that in each iteration, we have $f_{k+1} \leq f_k$, meaning that the value of the function decreases monotonically in each iteration. Therefore, it is possible for a sequence of iterations to move along a narrow valley on the curve, which usually occurs in difficult nonlinear problems. To overcome this problem, a valid alternative is to allow occasional increases in the objective function in some iterations. To easily understand the proposed algorithm, we first briefly recall the nonmonotone line-search technique GLL introduced by Grippo et al. [12].

Suppose that $\rho \in (0, 1)$, $\delta \in (0, 1)$, and M_0 is a nonnegative integer constant. The nonmonotone line search proposed in [12] determines the steplength α_k as:

$$\alpha_k = \rho^{j_k},$$

where j_k is the smallest nonnegative integer satisfying the following condition:

$$f(x_k + \alpha_k d_k) \leq \max_{0 \leq j \leq m(k)} \{f_{k-j}\} + \delta \alpha_k g_k^T d_k, \quad (6)$$

where

$$m(0) = 0, \quad 0 \leq m(k) \leq \min\{m(k-1) + 1, M_0\}.$$

If $m(k) = 0$, the above nonmonotone line search reduces to the Armijo line search. In nonmonotone line-search procedures, a controlled increase in the objective function value is permitted.

Many researchers have used nonmonotone line-search techniques [1, 7, 12, 13, 22, 28, 31]. Such methods not only raise the probability of locating a global optimum, but they can also enhance convergence speed in situations where a monotone approach is required to progress through a narrow valley. Although nonmonotone techniques that rely on (6) are effective in many cases, they possess certain shortcomings. First, a favorable function value produced at each iteration might be disregarded because of the maximum value present in (6). Second, in certain scenarios, the numerical outcomes are highly contingent upon the selection of M_0 (for further details, refer to [12, 27, 28]). Additionally, Dai [7] showed that if an iterative method produces iterations with R -linear convergence for a convex function, these iterations may not satisfy condition (6) for k sufficiently large, for any fixed value M_0 . To overcome these drawbacks, Zhang and Hager [31] put forward a more effective nonmonotone line-search method. This method shares a similar idea with the GLL line search, except that the term “max” is replaced by a weighted average of function values over successive iterations.

Specifically, their nonmonotone line search proceeds as follows:

$$f(x_k + \alpha_k d_k) \leq C_k + \sigma \alpha_k g_k^T d_k, \quad (7)$$

where

$$\begin{aligned} C_k &= \begin{cases} f_k, & \text{if } k = 0 \\ (\eta_{k-1}Q_{k-1}C_{k-1} + f_k)/Q_k, & \text{if } k \geq 1 \end{cases} \\ Q_k &= \begin{cases} 1, & \text{if } k = 0 \\ \eta_{k-1}Q_{k-1} + 1, & \text{if } k \geq 1, \end{cases} \end{aligned} \quad (8)$$

with $\eta_{k-1} \in [\eta_{min}, \eta_{max}]$ which η_{min} and η_{max} are two constants such that $0 \leq \eta_{min} \leq \eta_{max} < 1$. Numerical experiments indicate that nonmonotone line search (7) is practically more efficient than GLL line search [12].

The nonmonotone *BFGS* method was initially investigated by Liu et al. in [18]. Later, two other nonmonotone BFGS methods for solving problem (1) were introduced in [15, 19]. Notably, convergence analysis for all these algorithms has been proven under the assumption that the objective function is convex. In [3], Andrei, by scaling the third term in the standard *BFGS*, presented a scaled BFGS method with the aim of improving the self-correcting property of *BFGS*. Amini et al. in [1], by combining the algorithm proposed by Xiao in [29] and nonmonotone line search (7), presented a modified nonmonotone *BFGS* algorithm. They showed that this method has global convergence for nonconvex optimization problems.

In this paper, a new modified *BFGS* method is proposed based on the development of the idea [1, 3], which satisfies the sufficient decrease condition independently of line search. Numerical experiments demonstrate that the new method is effective and efficient.

This study is organized as follows. In the next section, the new algorithm is presented. The global convergence of the algorithm for general functions is analyzed in Section 3. In Section 4, we report some numerical experiments of the proposed method for unconstrained optimization problems, demonstrating the efficiency of the new algorithm. Finally, some conclusions are presented in the last section.

2 Motivation and algorithm

Two main approaches have been developed to improve the convergence properties of the standard BFGS algorithm, particularly for nonconvex optimization problems. The first approach is based on replacing the vector y_k in the secant condition with a modified vector y_k^* , thereby ensuring the positive definiteness of the Hessian approximation independently of the line-search procedure and without requiring convexity of the objective function. In contrast, the second approach focuses on introducing scaling parameters into the BFGS update formula to improve the spectral properties of the Hessian approximation and enhance its self-correcting property. Since these two approaches address different aspects of the BFGS algorithm, they can be regarded as complementary strategies whose integration may yield methods with stronger theoretical guarantees and improved numerical performance.

The positive definiteness of the updated matrix B_{k+1} in the standard BFGS method requires the condition [25]

$$s_k^T y_k > 0. \quad (9)$$

However, for nonconvex optimization problems, this condition is not necessarily satisfied under general line search procedures. Consequently, the updated matrix B_{k+1} may lose positive definiteness, and the resulting quasi-Newton direction may fail to be a descent direction.

To overcome this difficulty, several authors [1, 17, 29] have proposed replacing y_k by a corrected vector y_k^* such that

$$s_k^T y_k^* > 0. \quad (10)$$

A general form of such a correction is

$$y_k^* = y_k + t_k s_k, \quad t_k \geq 0, \quad (11)$$

where t_k is a nonnegative correction parameter. Different choices of t_k give rise to various modified BFGS methods with distinct convergence properties and numerical behavior.

To address the convergence difficulties of the standard BFGS method for nonconvex objective functions, Li and Fukushima [17] introduced the modified secant equation

$$B_{k+1} s_k = y_k^*, \quad (12)$$

where

$$y_k^* = y_k + t_k^* s_k, \quad (13)$$

with t_k^* defined as

$$t_k^* = \bar{C} \|g_k\|^\mu + \max \left\{ -\frac{s_k^T y_k}{\|s_k\|^2}, 0 \right\} \geq 0, \quad (14)$$

in which $\bar{C} > 0$ and $\mu > 0$ are given constants. With this definition, independently of the convexity of f and the line search employed, the following inequality holds:

$$s_k^T y_k^* \geq \bar{C} \|g_k\|^\mu \|s_k\|^2 > 0, \quad \forall k \in \mathbb{N}, \quad (15)$$

which guarantees the positive definiteness of B_{k+1} at every iteration. Li and Fukushima [17] established both global and superlinear convergence of the resulting algorithm for nonconvex optimization problems.

Subsequently, Xiao et al. [29] extended the framework proposed by Li and Fukushima by incorporating a nonmonotone line-search strategy. They established the global convergence of the resulting algorithm without requiring any convexity assumption on the objective function.

A second approach to improve the BFGS method is based on scaling the update formula rather than modifying the secant condition. As pointed out by Nocedal [24], the BFGS method corrects small eigenvalues of B_k more effectively than large ones, which may slow down the convergence of the algorithm. To overcome this drawback, scaled BFGS methods [3, 30], have been proposed, in which a positive scaling parameter γ_k is introduced into the third term of the standard update:

$$B_{k+1} = B_k - \frac{B_k s_k s_k^T B_k}{s_k^T B_k s_k} + \gamma_k \frac{y_k y_k^T}{s_k^T y_k}. \quad (16)$$

Yuan [30] proposed the choice $\gamma_k = \tilde{t}_k$, where $\tilde{t}_k$ is defined as

$$\tilde{t}_k = \frac{2}{s_k^T y_k} [f_k - f_{k+1} + s_k^T g_{k+1}], \quad (17)$$

and established global convergence and local superlinear convergence for convex objective functions.

A practically effective choice was subsequently proposed by Andrei [3], who defined

$$\gamma_k = \min \left\{ \frac{y_k^T s_k}{\|y_k\|^2 + \omega_k}, 1 \right\}. \quad (18)$$

Here, ω_k is a positive parameter, i.e., $\omega_k > 0$. Clearly, if $y_k^T s_k > 0$, then $0 < \gamma_k \leq 1$. In his numerical experiments, Andrei [3] uses $\omega_k = |s_k^T g_{k+1}|$.

As a result, the contribution of the third term is moderated, which helps control the growth of the eigenvalues of B_{k+1} and enhances the self-correcting property of the Hessian approximation. Moreover, if B_k is positive definite, then B_{k+1} remains positive definite. Andrei established global convergence of the resulting algorithm without assuming convexity of the objective function, and numerical experiments showed that it performs more efficiently than the standard BFGS method [3].

Following this line of research, Amini et al. [1] proposed a modified BFGS algorithm for unconstrained optimization in which B_k is updated according to

$$B_{k+1} = B_k - \frac{B_k s_k s_k^T B_k}{s_k^T B_k s_k} + \tau \frac{y_k^* (y_k^*)^T}{s_k^T y_k^*}, \quad (19)$$

where $\tau > 0$ is a scaling parameter and y_k^* is the Li and Fukushima correction vector, such that B_k satisfies the modified secant condition

$$B_{k+1} s_k = \tau y_k^*. \quad (20)$$

They established global convergence of the resulting algorithm for nonconvex optimization problems and reported encouraging numerical performance compared with both the standard BFGS method and the nonmonotone BFGS algorithm proposed by Xiao et al. [29].

In (19), proposed by Amini et al. [1], the parameter τ is treated as a fixed positive constant. On the other hand, Andrei [3] proposed an adaptive scaling parameter γ_k in place of a fixed constant; however, if $\gamma_k \rightarrow 0$, the corresponding rank-one term in the BFGS update is effectively lost. Motivated by combining these two ideas, we propose replacing the fixed parameter τ in (19) with the adaptive parameter γ_k of Andrei [3], leading to the update

$$B_{k+1} = B_k - \frac{B_k s_k s_k^T B_k}{s_k^T B_k s_k} + \gamma_k \frac{y_k^* (y_k^*)^T}{s_k^T y_k^*}. \quad (21)$$

However, this direct substitution reintroduces the same difficulty observed in [3]: when γ_k is small, the rank-one term $y_k^* (y_k^*)^T / (s_k^T y_k^*)$ is strongly attenuated, and if $\gamma_k \rightarrow 0$, the curvature information carried by y_k^* is effectively lost from the update. To overcome this issue, we bound γ_k from below by a positive constant τ , rather than using τ as a fixed value as in (19), and define the scaling parameter as

$$\gamma_k^* = \max\{\gamma_k, \tau\}. \quad (22)$$

This guarantees that the rank-one term never vanishes, while still allowing γ_k to shrink this term adaptively according to the local curvature. Unlike the fixed value τ used in [1], our choice adapts to the curvature information at each iteration while retaining the same safeguard. Based on this scaling parameter, we rewrite the BFGS update as

$$B_{k+1} = B_k - \frac{B_k s_k s_k^T B_k}{s_k^T B_k s_k} + \gamma_k^* \frac{y_k^* (y_k^*)^T}{s_k^T y_k^*}. \quad (23)$$

The proposed update combines the Li and Fukushima correction mechanism with an adaptive scaling strategy. As a result, it preserves the favorable convergence properties of modified secant methods while improving the spectral behavior of the Hessian approximation. Consequently, the proposed algorithm inherits both the strong convergence properties of modified secant methods and the favorable numerical behavior of scaled BFGS updates.

Moreover, it satisfies the following modified secant condition:

$$B_{k+1}s_k = \gamma_k^* y_k^*. \quad (24)$$

Remark 1. Clearly, setting $\gamma_k^* = 1$ in equation (24) yields the modified secant condition (20).

Remark 2. Since $\gamma_k^* > 0$ and inequality (15) holds, the matrix B_{k+1} generated by (23) remains positive definite.

Based on the proposed scaled modified BFGS update and the nonmonotone line-search strategy of Zhang and Hager, we now present the scaled modified BFGS algorithm for solving problem (1).

Algorithm 1: Nonmonotone Modified BFGS Algorithm

- 1: **Input:**
 - 2: Choose an initial point $x_0 \in \mathbf{R}^n$ and an initial symmetric positive definite matrix $B_0 \in \mathbf{R}^{n \times n}$.
 - 3: Specify the constants $\sigma, \rho \in (0, 1)$ together with the positive constants $\bar{C}, \mu$, and a sufficiently small $\varepsilon > 0$.
 - 4: **Step 0.** Set $Q_0 = 1, C_0 = f_0$ and $k = 0$.
 - 5: **Step 1.** Terminate the algorithm when $\|g_k\| \leq \varepsilon \|g_0\|$.
 - 6: **Step 2.** Compute the search direction d_k by solving $B_k d_k = -g_k$.
 - 7: **Step 3.** Set $\alpha_k = \rho^{j_k}$, where j_k denotes the smallest non-negative integer for which α_k satisfies (7).
 - 8: **Step 4.** Update the iterate by $x_{k+1} = x_k + \alpha_k d_k$.
 - 9: **Step 5.** Compute B_{k+1} using (23), where y_k^* is defined by (13).
 - 10: **Step 6.** Set $k = k + 1$ and go to Step 1.
-

The next section establishes the global convergence of the proposed Algorithm 1 for nonconvex unconstrained optimization problems.

3 Convergence analysis

In this section, we prove the global convergence of the proposed algorithm under the following assumptions, which hold throughout this section.

(i) The level set

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

is bounded.

(ii) The function f is continuously differentiable on a neighborhood L_0 containing $L(x_0)$, and its gradient g is Lipschitz continuous on L_0 . That is, there exists a constant $L > 0$ such that

$$\|g(x) - g(y)\| \leq L\|x - y\|, \quad \forall x, y \in L_0,$$

where $\|\cdot\|$ denotes the Euclidean norm.

The proof proceeds through a sequence of lemmas: Lemma 1 shows that the nonmonotone line search is well-defined and $f_k \leq C_k$; Lemma 2 shows that $\{C_k\}$ is nonincreasing and the iterates remain within a bounded level set; Lemma 3 establishes uniform bounds on B_k along the directions s_k ; Lemma 4 gives a lower bound on the steplength; and Lemma 5 establishes a summability result. These results are then combined to prove the main convergence theorem.

Lemma 1. *Assume that the sequence $\{x_k\}$ is produced by Algorithm 1. Then, $f_k \leq C_k$ for all $k \in \mathbb{N} \cup \{0\}$. Moreover, the nonmonotone line search (7) is well defined.*

Proof. Since $f(x)$ is continuous and Assumption (i) holds, $f(x)$ is bounded, and since $g_k^T d_k \leq 0$, the result follows by the same argument as in Lemma 1.1 of [31]. $\square$

Lemma 2. *Assume that the sequence $\{x_k\}$ is produced by Algorithm 1. Then, $\{C_k\}$ is a nonincreasing sequence and for all $k \in \mathbb{N} \cup \{0\}$*

$$\{x_k\} \subset L(x_0). \quad (25)$$

Proof. Since B_k is a positive definite matrix, we obtain

$$g_k^T d_k = -d_k^T B_k d_k < 0.$$

So

$$f(x_k + \alpha_k d_k) \leq C_k + \delta \alpha_k g_k^T d_k \leq C_k,$$

as a result $f_{k+1} \leq C_k$ and C_{k+1} is a convex combination of C_k and $f(x_{k+1})$, we have $C_{k+1} \leq C_k$ for each k . Hence

$$f_{k+1} \leq C_k \leq C_{k-1} \leq \dots \leq C_0 = f(x_0),$$

which implies that all the iterates x_k are contained in the level set

$$L = \{x \in \mathbb{R}^n | f(x) \leq f(x_0)\}. \quad \square$$

Lemma 3. *Let the sequence $\{x_k\}$ be generated by Algorithm 1. If $\|g_k\| \geq \zeta$ holds for all k with some constant $\zeta > 0$, then there exist positive constants β_j , $j = 1, 2, 3$, such that, for all $k \in \mathbb{N}$, the inequalities*

$$\|B_i s_i\| \leq \beta_1 \|s_i\|, \quad \beta_2 \|s_i\|^2 \leq s_i^T B_i s_i \leq \beta_3 \|s_i\|^2, \quad (26)$$

hold for at least a half of the indices $i \in \{1, 2, \dots, k\}$.

Proof. Since the modified vector y_k^* retains the same Li and Fukushima type structure as in [1], the relations

$$\frac{y_k^{*T} s_k}{\|s_k\|^2} \geq m_1, \quad \frac{\|y_k^*\|^2}{y_k^{*T} s_k} \leq M_1$$

follow by the same argument as in Lemma 3.3 of [1]. $\square$

Lemma 4. *Suppose the sequence $\{x_k\}$ is generated by Algorithm 1. If, for all k , $\|g_k\| \geq \zeta$ holds for some constant $\zeta > 0$, then a positive constant $\bar{\alpha}$ exists such that $\alpha_j > \bar{\alpha}$ for every $j \in J = \{j \mid (26) \text{ holds}\}$.*

Proof. It is enough to consider the case $\alpha_j \neq 1$. Since backtracking is used, $\hat{\alpha}_k \equiv \alpha_k/\rho$ denotes the trial steplength that was rejected by the nonmonotone line search (7), while α_k is the accepted steplength. Since $f_k \leq C_k$, we have

$$f(x_k + \hat{\alpha}_k d_k) - f_k > \sigma \hat{\alpha}_k g_k^T d_k.$$

By the mean-value theorem, for some $\theta \in (0, 1)$,

$$g(x_k + \theta \hat{\alpha}_k d_k)^T d_k > \sigma g_k^T d_k. \quad (27)$$

Combining (27) with the Cauchy–Schwarz inequality, Assumption (ii), and the lower bound β_2 from Lemma 3, we obtain

$$\hat{\alpha}_k L \|d_k\|^2 \geq (1 - \sigma) d_k^T B_k d_k \geq (1 - \sigma) \beta_2 \|d_k\|^2,$$

so that $\hat{\alpha}_k > \beta_2(1 - \sigma)/L$, which implies $\alpha_k > \bar{\alpha}$ for every $k \in J$, where $\bar{\alpha} = \beta_2 \rho(1 - \sigma)/L$ is a positive constant. $\square$

Lemma 5. *Let the sequence $\{x_k\}$ be generated by Algorithm 1, then*

$$\sum_{k=0}^{\infty} (-g_k^T s_k) < \infty. \quad (28)$$

Proof. The relations (7) and (8) result that

$$C_{k+1} = \frac{\eta_k Q_k C_k + f_{k+1}}{Q_{k+1}} \leq \frac{(\eta_k Q_k + 1)C_k}{Q_{k+1}} + \frac{\sigma \alpha_k g_k^T d_k}{Q_{k+1}} = C_k + \frac{\sigma \alpha_k g_k^T d_k}{Q_{k+1}}.$$

While the last equality follows from $Q_{k+1} = \eta_k Q_k + 1$, we conclude that

$$C_k - C_{k+1} \geq \frac{-\sigma \alpha_k g_k^T d_k}{Q_{k+1}}. \quad (29)$$

Furthermore, it has been established in [31] that

$$Q_{k+1} = 1 + \sum_{j=0}^k \prod_{i=0}^j \eta_{k-i} \leq 1 + \sum_{j=0}^k \eta_{\max}^{j+1} \leq \sum_{j=0}^{\infty} \eta_{\max}^j = \frac{1}{1 - \eta_{\max}}. \quad (30)$$

Substituting (30) into (29) concludes that

$$C_k - C_{k+1} \geq \sigma(1 - \eta_{\max})(-\alpha_k g_k^T d_k),$$

therefore

$$\sum_{k=0}^{\infty} (C_k - C_{k+1}) \geq \sigma(1 - \eta_{\max}) \sum_{k=0}^{\infty} (-\alpha_k g_k^T d_k). \quad (31)$$

By Lemma 2, $\{C_k\}$ is nonincreasing, and since $\{x_k\} \subset L(x_0)$ with f bounded below on $L(x_0)$ (Assumption (i)), $\{C_k\}$ is also bounded below. Hence $\lim_{k \rightarrow \infty} C_k$ exists, and it follows from (31) that

$$\sigma(1 - \eta_{\max}) \sum_{k=0}^{\infty} (-\alpha_k g_k^T d_k) \leq C_0 - \lim_{k \rightarrow \infty} C_k < \infty.$$

So (28) holds and the proof is completed. $\square$

Now, the main result of this section, the global convergence of the proposed algorithm with non-monotone line search for general functions, can be described.

Theorem 1. *Suppose that Assumptions (i) and (ii) are satisfied then the sequence $\{x_k\}$ generated by Algorithm 1, is globally convergent, i.e.*

$$\liminf_{k \rightarrow \infty} \|g_k\| = 0. \quad (32)$$

Proof. The proof is by contradiction. Suppose that (32) does not hold. So, there exists a constant $\zeta > 0$ such that

$$\|g_k\| \geq \zeta,$$

for all k is sufficiently large. It follows from (28) and $B_k s_k = \alpha_k B_k d_k = -\alpha_k g_k$, that

$$\sum_{k=0}^{\infty} \alpha_k \frac{s_k^T B_k s_k}{\|B_k s_k\|^2} \|g_k\|^2 = \sum_{k=0}^{\infty} \frac{1}{\alpha_k} s_k^T B_k s_k = \sum_{k=0}^{\infty} (-\alpha_k g_k^T d_k) < \infty.$$

Together with $\|g_k\| \geq \zeta$ and definition of J in Lemma 4, we obtain

$$\sum_{k=0}^{\infty} \alpha_k \frac{s_k^T B_k s_k}{\|B_k s_k\|^2} \|g_k\|^2 \geq \zeta^2 \sum_{k \in J} \alpha_k \frac{s_k^T B_k s_k}{\|B_k s_k\|^2} > \zeta^2 \bar{\alpha} \sum_{k \in J} \frac{s_k^T B_k s_k}{\|B_k s_k\|^2}.$$

The last inequality follows from Lemma 4. This implies

$$\sum_{k \in J} \frac{s_k^T B_k s_k}{\|B_k s_k\|^2} < \infty. \quad (33)$$

Since the set J is infinite, it is concluded that $\frac{s_k^T B_k s_k}{\|B_k s_k\|^2} \rightarrow 0$ for $k \in J$. This immediately contradicts the fact

$$\frac{s_k^T B_k s_k}{\|B_k s_k\|^2} \geq \frac{\beta_2 \|s_k\|^2}{\beta_1^2 \|s_k\|^2} = \frac{\beta_2}{\beta_1^2},$$

that is an obvious result of (26). Thus this completes the proof. $\square$

4 Numerical experiments

This section evaluates the numerical performance of the proposed modified nonmonotone BFGS algorithm (MBFGS-NH) on 76 large-scale problems taken from Andrei [2] and Moré et al. [23] with dimensions $n \geq 100$ (ranging from 100 to 1000). Table 1 presents the list of test problems together with their dimensions. Since the primary focus of this study is on large-scale optimization, small-scale problems (with $n < 100$) were excluded from the benchmark set to ensure a fair evaluation of algorithmic scalability. To assess the new algorithm, we compare it with the standard BFGS method, denoted as BFGS, the modified BFGS algorithm of Amini et al., denoted as MBFGS-BH [1], the modified BFGS method of Andrei with the Hager line search, denoted as MBFGS-AH [3], and the recent modified BFGS method of Hassan and Moghrabi, denoted as MBFGS-HM [14].

For each test problem, all algorithms start from the same initial point, with the initial Hessian approximation set to $B_0 = I$.

The termination criterion is defined as

$$\|g_k\| \leq 10^{-6} \|g_0\|.$$

If the number of iterations exceeds 5000, the algorithm is terminated and the corresponding result is reported as “inf” in the tables. Such cases are considered unsuccessful runs. In the performance profiles, unsuccessful runs are assigned a sufficiently large value M instead of infinity, following the standard convention of Dolan and Moré [9]. Specifically, for each algorithm, the value of M is set to ten times the largest finite value obtained by that algorithm over the entire test set. This ensures that unsuccessful runs are placed far to the right of the performance profiles and, consequently, the average values reported in the summary table are computed using this failure penalty over all test problems.

All algorithms were implemented in MATLAB R2024a. The parameters used in the line searches are selected according to the settings commonly adopted in related studies; see, for example, [1, 3]. Specifically, we set

$$\sigma = 10^{-4}, \quad \rho = 0.8, \quad \eta_k = 0.2.$$

Moreover, for the scaling parameter γ_k^* in (22), we set

$$\tau_k = 0.1,$$

following Amini et al. [1]. For the parameter ω_k , we follow Andrei [3] and select

$$\omega_k = |s_k^T g_{k+1}|.$$

The parameters in the modified secant condition are chosen as $\mu = 4$, and $\bar{C}$ is defined as follows:

$$\bar{C} = \begin{cases} 10^{-2}, & \text{if } \|g_k\| \leq 10^{-2}, \\ 0, & \text{otherwise.} \end{cases}$$

The following notations are used in these tables:

No: the number of the tested problem.

Problem: the name of the test problem.

Dim: the dimension of the problem.

Iter: the number of iterations.

Numf: the number of function evaluations.

Time: the CPU time in seconds.

The detailed numerical results, including the number of iterations, function evaluations, and CPU time, are reported in Tables 2 and 3. Due to the large number of test problems, the results are divided into two tables for better readability. For all tested algorithms, the number of iterations coincides with the number of gradient evaluations; therefore, the iteration counts also represent the number of gradient evaluations.

When comparing two or more algorithms based on numerical results, the superior algorithm may not be obvious from the results presented. To facilitate a meaningful comparison between algorithms,

Table 1: Test problems and their dimensions

No	Problem	Dim	No	Problem	Dim
1	Almost Perturbed Quadratic	1000	39	Extended Powell Singular	1000
2	ARGLINB	1000	40	Extended PSC1	1000
3	ARWHEAD	1000	41	Extended quadratic exponential EP1	1000
4	BDEXP	1000	42	Extended quadratic penalty QP1	1000
5	BDQRTIC	1000	43	Extended quadratic penalty QP2	1000
6	BG2	1000	44	Extended Rosenbrock	1000
7	BIGGSB1	1000	45	Extended TET	1000
8	Broyden Tridiagonal	1000	46	Extended Tridiagonal 1	1000
9	CUBE	1000	47	Extended Tridiagonal 2	1000
10	Diagonal 2	1000	48	Extended Trigonometric	200
11	Diagonal 4	1000	49	Extended Wood	1000
12	Diagonal 5	1000	50	Fletcher	1000
13	Diagonal 7	1000	51	Full Hessian FH3	1000
14	Diagonal 8	1000	52	Generalized Tridiagonal 1	100
15	DIXMAANA	999	53	Generalized PSC1	1000
16	DIXMAANB	999	54	Hager	1000
17	DIXMAANC	999	55	HIMMELBG	1000
18	DIXMAAND	999	56	HIMMELH	1000
19	DIXMAANE	999	57	LIARWHD	100
20	DIXMAANF	999	58	LIARWHD	1000
21	DIXMAANG	999	59	NONDQUAR	1000
22	DIXMAANH	999	60	NONSCOMP	1000
23	DIXMAANI	999	61	Partial Perturbed Quadratic	1000
24	DIXMAANJ	999	62	Penalty I	500
25	DIXMAANK	999	63	Penalty II	100
26	DIXMAANL	999	64	Perturbed Quadratic diagonal	1000
27	DIXON3DQ	1000	65	Perturbed Quadratic	1000
28	DQDRTIC	1000	66	POWER	1000
29	EDENSCH	1000	67	Perturbed Tridiagonal quadratic	1000
30	ENGVAL1	1000	68	Quadratic QF1	1000
31	Extended BD1	1000	69	Quadratic QF2	1000
32	Extended Beale	1000	70	QUARTC	1000
33	Extended Cliff	1000	71	Raydan 2	1000
34	Extended DENSCHNF	1000	72	SINCOS	1000
35	Extended Freudenstein and Roth	1000	73	Staircase 1	1000
36	Extended Himmelblau	1000	74	Staircase 2	1000
37	Extended Penalty	1000	75	TRIDIA	1000
38	Extended Powell	1000	76	Variably Dimensioned	100

we use the performance profiles proposed by Dolan and Moré [9]. In these profiles, the fraction P of the problems for which a method is within a factor τ of the best time is plotted. The left side of the figure gives the percentage of the test problems for which a method is the fastest; the right side gives the percentage of the test problems that were successfully solved by each of the methods. The top curve is the method that solved the most problems in a time that was within a factor τ of the best time. With this tool, in Figures 1, we compare the performance of each algorithm based on Iter, Numf, and Time for 76 large-scale test problems, respectively.

A closer inspection of these figures yields the following observations.

Figure 1(a) indicates that Andrei-Hager uses the fewest iterations on approximately 31.58% of the test problems, while New-Hager and BFGS solve almost 28.95% and 27.63% of the test problems, respectively. However, for larger τ , New-Hager becomes the top curve and reaches 90.79% at $\tau = 10$.

Figure 1(b) shows that New-Hager solves almost 35.53% of the test problems with the least number of function evaluations, while Hassan-Moghrabi and Bahrami-Hager solve almost 34.21% and 25.00% of the test problems, respectively. New-Hager remains the top curve over the whole range of τ and reaches 88.16% at $\tau = 10$.

Figure 1(c) shows that BFGS and Hassan-Moghrabi solve almost 26.32% of the test problems with the least CPU time, while Andrei-Hager and New-Hager solve almost 22.37% and 19.74% of the test

Table 2: Numerical results of Iter, Numf, and CPU time

No	MBFGS-NH	MBFGS-BH	MBFGS-AH	BFGS	MBFGS-HM
	Iter/Numf/Time	Iter/Numf/Time	Iter/Numf/Time	Iter/Numf/Time	Iter/Numf/Time
1	672/17465/257.8906	672/17465/254.5156	370/11465/151.2344	2223/73232/1319.4063	1689/17586/435.3125
2	74/1192/28.5000	74/1192/26.5625	327/57880/207.5313	89/8707/43.0313	3/64/1.0156
3	86/761/29.0000	86/761/29.4219	161/5667/55.0469	119/3944/36.1406	834/9682/299.0156
4	5/6/2.0313	7/8/2.3594	5/6/1.6094	17/35/5.2031	17/35/6.0781
5	311/4960/117.6563	311/4960/126.0313	397/16980/205.9063	1421/56656/492.3438	1240/19051/362.0781
6	120/2356/42.2656	120/2356/42.2969	inf/inf/inf	inf/inf/inf	inf/inf/inf
7	1540/8553/729.0469	1851/14387/941.3125	1540/8544/806.1094	inf/inf/inf	1878/5211/439.8594
8	384/5773/141.6406	384/5773/139.7656	461/9204/171.7188	412/8797/133.0781	311/2032/75.0938
9	690/8806/225.5781	690/8806/225.4375	342/9698/115.3281	4236/124470/1416.4375	730/8103/162.5156
10	190/258/70.5156	1094/8581/425.5469	193/265/70.3281	2659/5319/916.2656	436/1210/102.8281
11	90/837/36.9219	90/837/37.0000	82/1543/33.7031	2/23/0.6719	322/2536/82.1563
12	10/11/3.9688	90/760/35.6250	10/11/3.9688	5/11/1.7500	6/13/1.2656
13	397/1591/162.1250	9/11/3.5625	397/1591/160.1094	7/16/2.4844	6/15/1.4844
14	837/3355/340.2031	93/787/37.8594	837/3355/341.8438	7/19/2.2813	9/26/2.1719
15	65/163/27.2500	93/785/36.7344	65/163/26.3594	12/30/3.7656	113/359/24.9063
16	89/201/38.3281	93/785/38.5625	89/201/35.2969	13/28/4.0781	111/329/25.1094
17	67/168/29.3594	93/785/38.0781	67/168/29.7500	24/50/8.2031	160/763/36.5781
18	68/183/27.3125	89/752/35.4844	68/183/26.7188	26/55/8.2188	156/497/34.9688
19	183/212/75.0938	866/6759/353.9219	183/212/75.0938	380/761/127.2344	284/819/67.3906
20	178/211/76.9219	740/5836/312.5781	178/211/76.9219	169/339/59.6250	389/1866/89.2656
21	183/218/79.6094	720/5677/307.5625	183/218/79.6094	216/433/76.5469	410/1474/92.8125
22	153/184/67.5000	671/5287/290.8594	153/184/67.5000	245/493/84.7500	524/2695/120.0938
23	1076/1161/458.0625	inf/inf/inf	1054/1132/391.1875	1457/2915/472.5781	inf/inf/inf
24	362/405/144.4844	2311/17936/862.6250	389/431/147.6406	266/533/82.6250	704/2143/159.2188
25	306/338/117.4375	2051/15976/751.2344	330/373/127.6094	398/797/123.8125	1038/5406/232.4688
26	256/289/98.7031	1225/9509/448.9688	253/286/95.5938	246/493/75.3906	536/3630/146.7500
27	1562/8849/502.0313	1850/14377/596.4844	1562/8849/504.2344	inf/inf/inf	2267/6956/600.6094
28	92/987/36.0469	92/987/38.5156	144/3588/62.4375	414/9855/148.8125	708/9375/187.5781
29	90/854/29.9063	90/854/30.3594	84/925/27.9844	83/1158/24.3438	94/597/23.7500
30	123/1103/40.5781	123/1103/40.5000	83/965/26.9219	153/1993/43.1094	53/382/13.3594
31	109/909/38.9219	108/918/38.8125	64/686/22.8750	92/888/27.3281	62/241/17.4219
32	108/985/33.4531	108/985/34.4531	92/1325/42.0938	424/6758/162.8594	664/4741/155.1250
33	1/81/0.4063	1/81/0.4219	1/81/0.2969	1/82/0.1875	1/25/0.3281
34	126/2006/41.8281	126/2006/42.3281	68/1614/22.6406	116/2749/34.3438	30/268/7.3281
35	133/2792/48.1406	133/2792/53.3438	150/5238/58.0781	4458/159152/1422.2188	3915/41412/948.6875
36	94/1005/38.7344	49/1005/38.1094	184/3665/75.2969	131/2580/45.7344	42/253/10.2813
37	51/1663/19.4688	51/1663/17.7344	31/2208/11.2813	28/1812/8.0000	11/215/2.5469
38	278/2996/117.4844	278/2996/112.1563	161/3473/71.2344	inf/inf/inf	1135/5142/293.0625

problems, respectively. For larger values of τ , New-Hager becomes the leading method in the profile, reaching 89.47% at $\tau = 10$.

The aggregate numerical results are summarized in Table 4. New-Hager achieves the lowest average number of iterations (327.08), function evaluations (4983.30), and CPU time (28.35 seconds) among all methods. It is also the only method that successfully solves all 76 test problems (100% success rate), whereas the standard BFGS method fails on 10 problems. These results confirm the superior efficiency and robustness of the proposed New-Hager method for large-scale optimization.

5 Conclusions

In this paper, we proposed a modified nonmonotone BFGS algorithm, denoted as MBFGS-NH, for solving nonconvex unconstrained optimization problems. The method combines a modified BFGS update with the Hager–Zhang nonmonotone line search, which allows occasional increases in the objective function at some iterations. Under standard assumptions, we established global convergence to a stationary

Table 3: Numerical results of Iter, Numf, and CPU time (continued)

No	MBFGS-NH	MBFGS-BH	MBFGS-AH	BFGS	MBFGS-HM
	Iter/Numf/Time	Iter/Numf/Time	Iter/Numf/Time	Iter/Numf/Time	Iter/Numf/Time
39	249/2704/87.0156	249/2704/85.1719	182/3929/68.7656	inf/inf/inf	1421/11874/575.6719
40	70/321/29.6406	67/569/28.3125	424/2577/175.1563	77/418/27.6875	50/286/11.3125
41	90/1153/32.9063	90/1153/32.8125	50/1300/18.0000	4/34/1.1875	529/12485/123.0000
42	61/558/23.8125	59/544/22.1719	49/734/18.6875	45/607/15.0938	57/244/22.5625
43	239/2022/93.2656	229/1952/85.7969	242/6703/106.2188	47/1086/17.2031	45/138/17.0000
44	354/6321/126.6563	354/6321/123.5469	110/3297/39.6094	inf/inf/inf	1241/6813/511.7813
45	95/807/40.3438	95/807/40.9531	1084/9764/446.3906	120/1058/41.2500	87/653/20.9063
46	120/566/49.9375	82/693/35.3594	120/566/50.5156	9/38/3.0000	184/553/45.2344
47	265/1859/96.7656	137/1140/50.2969	265/1859/97.8281	94/687/29.0313	1282/8354/292.9688
48	193/480/2.4375	430/3459/4.5156	130/256/1.1250	72/145/0.6563	599/1783/5.3438
49	1596/26696/638.7969	1596/26696/659.6563	192/5662/89.5156	inf/inf/inf	1779/13829/446.8906
50	2060/37245/886.1250	2060/37245/799.9063	4110/128414/1642.9375	inf/inf/inf	inf/inf/inf
51	94/824/31.7188	94/824/33.1875	729/23329/263.1719	3/38/1.1563	4/19/0.8750
52	109/1169/0.6250	103/1110/0.6406	83/1173/0.4531	106/1697/0.5625	79/563/0.3750
53	73/631/30.6406	221/1993/90.9375	19/155/7.7344	60/873/21.7656	87/401/20.4375
54	250/2278/109.7813	250/2278/104.9844	103/1327/44.1406	126/1818/44.2188	244/2584/60.0469
55	2/3/0.7813	2/3/0.6250	2/3/0.6719	2/5/0.6250	19/41/4.7344
56	7/8/2.0625	7/8/1.8438	10/11/2.7344	10/21/2.7031	4/186/0.7188
57	85/813/0.3125	85/813/0.2969	105/3020/0.3125	70/1761/0.2344	985/8662/3.4844
58	120/1370/35.9844	120/1370/35.2813	120/4656/41.2656	83/2856/24.9531	1337/17084/495.4531
59	311/1368/126.0781	399/3188/150.6875	438/2584/175.8281	1208/7019/446.7656	783/2405/196.5000
60	708/8608/245.1563	708/8608/249.7656	186/2945/73.4531	3226/53088/1138.6719	858/4491/317.3281
61	579/15084/189.9688	579/15084/191.8594	124/4787/42.2500	130/4794/38.8125	363/2242/89.4375
62	46/1249/3.7344	46/1249/2.4219	26/1646/1.6406	30/1552/2.4531	9/158/0.2813
63	120/4851/1.6875	120/4851/1.2031	44/1513/0.4531	27/1025/0.1875	25/391/0.2656
64	85/750/34.0938	85/750/34.2813	1416/45313/559.9063	151/4387/49.8906	1370/12832/316.0469
65	672/17466/274.2656	672/17466/267.2656	426/13225/180.2344	2158/71105/761.4844	1209/14621/353.1563
66	1082/55113/472.3281	1082/55113/421.3594	3796/231074/1265.0625	inf/inf/inf	inf/inf/inf
67	689/17901/279.5625	689/17901/261.9844	433/13461/214.3125	2249/74156/1443.3906	1130/10198/278.5156
68	488/11236/183.7031	488/11236/219.2500	394/10971/169.5156	2223/66310/1346.7344	1493/16176/455.0313
69	854/24819/307.7344	854/24819/310.4375	531/17948/246.8281	3982/143543/1515.7188	1233/16359/329.4844
70	10/15/3.3281	28/212/7.8906	14/22/4.0938	15/35/3.7656	1/5/0.2909
71	10/11/3.1719	91/765/8.9219	10/11/4.3125	7/15/4.5000	9/19/1.8281
72	70/321/3.6250	67/569/33.8594	424/2577/3.5781	77/418/2.1563	50/286/12.5625
73	117/2038/26.5313	117/2038/24.7188	581/33755/162.2344	4792/268248/24.6563	inf/inf/inf
74	120/2069/37.8281	120/2069/36.4688	528/30349/227.3594	4461/249732/1590.7031	inf/inf/inf
75	977/32036/40.4219	977/32036/38.3281	898/36795/194.5625	inf/inf/inf	4069/49910/1171.2500
76	25/319/0.0156	25/319/0.1094	32/3740/0.1094	11/762/0.1094	22/183/0.1094

Table 4: Aggregate numerical results over the 76 large-scale test problems

Method	Solved	Failures	Avg Iter	Avg Numf	Avg Time (s)
MBFGS-NH	76	0	327.08	4983.30	28.35
MBFGS-BH	75	1	724.68	13383.47	65.11
MBFGS-AH	75	1	906.13	41786.93	77.03
BFGS	66	10	6586.76	353863.14	896.70
MBFGS-HM	70	6	3785.33	44297.11	995.62

point without requiring convexity of the objective function.

The numerical experiments were conducted on 76 large-scale test problems with dimensions $n \geq 100$. MBFGS-NH solved all 76 test problems, achieving a 100% success rate, whereas the standard BFGS method failed on 10 problems. Moreover, MBFGS-NH achieves the lowest average number of

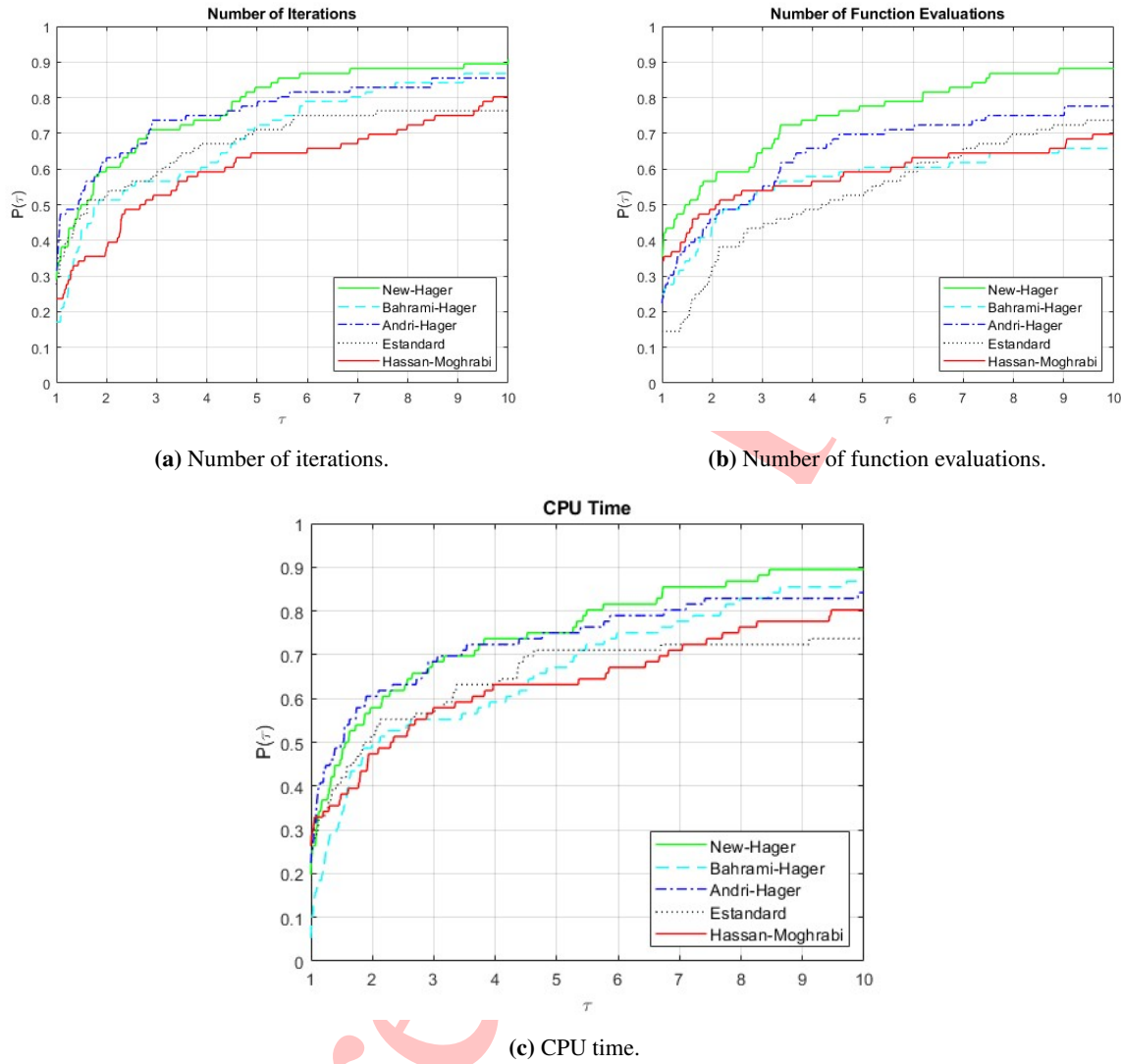


Figure 1: Performance profiles of the tested algorithms

iterations (327.08), function evaluations (4983.30), and CPU time (28.35 seconds) compared with the standard BFGS method and the modified BFGS methods of Amini et al. [1], Andrei [3], and Hassan-Moghrabi [14]. The performance profiles further confirmed that MBFGS-NH consistently ranks among the top two methods for $\tau \geq 2$ in all three metrics, demonstrating its superior efficiency and robustness for large-scale optimization.

References

- [1] K. Amini, S. Bahrami, S. Amiri, A nonmonotone modified BFGS algorithm for nonconvex unconstrained optimization problems, *Filomat* **30**(5) (2016) 1283–1296.

- [2] N. Andrei, An unconstrained optimization test functions collection, *Adv. Model. Optim.* **10**(1) (2008) 147–161.
- [3] N. Andrei, An adaptive scaled BFGS method for unconstrained optimization, *Numer. Algorithms* **77**(2) (2017) 413–432.
- [4] R.H. Byrd, J. Nocedal, A tool for the analysis of quasi-Newton methods with application to unconstrained minimization, *SIAM J. Numer. Anal.* **26** (1989) 727–739.
- [5] R.H. Byrd, J. Nocedal, Y. Yuan, Global convergence of a class of quasi-Newton methods on convex problems, *SIAM J. Numer. Anal.* **24** (1987) 1171–1189.
- [6] Y.H. Dai, Convergence properties of the BFGS algorithm, *SIAM J. Optim.* **13** (2002) 693–701.
- [7] Y.H. Dai, On the nonmonotone line search, *J. Optim. Theory Appl.* **112** (2002) 315–330.
- [8] R. Dehghani, N. Bidabadi, M.M. Hosseini, Using nonlinear functions to approximate a new quasi-Newton method for unconstrained optimization problems, *Numer. Algor.* **87** (2021) 755–777.
- [9] E.D. Dolan, J.J. Moré, Benchmarking optimization software with performance profiles, *Math. Program.* **91** (2002) 201–213.
- [10] R. Fletcher, Practical Methods of Optimization: Volume 1, Unconstrained Optimization, John Wiley and Sons, Chichester, 1980.
- [11] P.E. Gill, J.H. Runnoe, On recent developments in BFGS methods for unconstrained optimization, Technical Report CCoM-22-4, UCSD Center for Computational Mathematics, 2023.
- [12] L. Grippo, F. Lampariello, S. Lucidi, A nonmonotone line search technique for Newton’s method, *SIAM J. Numer. Anal.* **23** (1986) 707–716.
- [13] C. Gu, D. Zhu, A nonmonotone line search filter method with reduced Hessian updating for nonlinear optimization, *J. Syst. Sci. Complex.* **26** (2013) 534–555.
- [14] B.A. Hassan and I.A.R. Moghrabi, A modified secant equation quasi-Newton method for unconstrained optimization, *J. Appl. Math. Comput.* **69** (2023) 451–464.
- [15] J. Han, G. Liu, Global convergence analysis of a new nonmonotone BFGS algorithm on convex objective function, *Comput. Optim. Appl.* **7** (1997) 277–289.
- [16] D.H. Li, M. Fukushima, A modified BFGS method and its global convergence in nonconvex minimization, *J. Comput. Appl. Math.* **129** (2001) 15–35.
- [17] D.H. L, M. Fukushima, On the global convergence of the BFGS method for nonconvex unconstrained optimization problems, *SIAM J. Optim.* **11**(4) (2001) 1054–1064.
- [18] G. Liu, J. Han, D. Sun, Global convergence of the BFGS algorithm with nonmonotone line search, *Optimization* **34** (1995) 147–159.

- [19] L. Liu, S. Yao, Z. Wei, The global and superlinear convergence of a new nonmonotone MBFGS algorithm on convex objective function, *J. Comput. Appl. Math.* **220** (2008) 422–438.
- [20] F. Mannel, A Globalization of L-BFGS and the Barzilai-Borwein method for nonconvex unconstrained optimization, *J. Optim. Theory Appl.* **204(3)** (2025) 1–34.
- [21] W.F. Mascarenhas, The BFGS method with exact line searches fails for non-convex objective functions, *Math. Program.* **99** (2004) 49–61.
- [22] M. Miladinovic, P. Stanimirović, S. Miljković, Scalar Correction Method for Solving Large-Scale Unconstrained Minimization Problems, *J. Optim. Theory Appl.* **151** (2011) 304–320.
- [23] J.J. Moré, B.S. Garbow, K.E. Hillstom, Testing unconstrained optimization software, *ACM Trans. Math. Softw.* **7(1)** (1981) 17–41.
- [24] J. Nocedal, Y. Yuan, Analysis of self-scaling quasi-Newton method, *Math. Program.* **61** (1993) 19–37.
- [25] J. Nocedal, S.J. Wright, *Numerical Optimization*, Springer, New York, 2006.
- [26] M.J.D. Powell, Some global convergence properties of a variable metric algorithm for minimization without exact line searches, in: R.W. Cottle and C.E. Lemke (Eds.), *Nonlinear Programming, SIAM–AMS Proceedings, Vol. LX*, American Mathematical Society, Providence, 1976, pp. 53–72.
- [27] M. Raydan, The Barzilai and Borwein gradient method for the large-scale unconstrained minimization problem, *SIAM J. Optim.* **7** (1997) 26–33.
- [28] Ph.L. Toint, An assessment of nonmonotone linesearch technique for unconstrained optimization, *SIAM J. Sci. Comput.* **17**(1996)725 – –739.
- [29] Y. Xiao, H. Sun, Z. Wang, A globally convergent BFGS method with nonmonotone line search for non-convex minimization, *J. Comput. Appl. Math.* **230** (2009) 95–106.
- [30] Y. Yuan, A modified BFGS algorithm for unconstrained optimization, *IMA J. Numer. Anal.* **11** (1991) 325–332.
- [31] H.C. Zhang, W.W. Hager, A nonmonotone line search technique and its application to unconstrained optimization, *SIAM J. Optim.* **14** (2004) 1043–1056.