Filomat 30:5 (2016), 1283–1296
DOI 10.2298/FIL1605283A

Published by Faculty of Sciences and Mathematics,
University of Niš, Serbia
Available at: http://www.pmf.ni.ac.rs/filomat

A Nonmonotone Modified BFGS Algorithm for Nonconvex Unconstrained Optimization Problems

Keyvan Amini[a], Somayeh Bahrami[a], Shadi Amiri[a]

[a]*Department of Mathematics, Faculty of Sciences, Razi University, Kermanshah, Iran.*

Abstract. In this paper, a modified BFGS algorithm is proposed to solve unconstrained optimization problems. First, based on a modified secant condition, an update formula is recommended to approximate Hessian matrix. Then thanks to the remarkable nonmonotone line search properties, an appropriate nonmonotone idea is employed. Under some mild conditions, the global convergence properties of the algorithm are established without convexity assumption on the objective function. Preliminary numerical experiments are also reported which indicate the promising behavior of the new algorithm.

1. Introduction

In this paper, we consider the unconstrained optimization problem

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

where $f : \mathbb{R}^n \to \mathbb{R}$ is a continuously differentiable function. Iterative methods are usually used for solving this problem by generating a sequence $\{x_k\}$ as follows:

$$x_{k+1} = x_k + \alpha_k d_k, \tag{2}$$

for $k \geq 0$, where d_k is a search direction, $\alpha_k > 0$ is a steplength and x_0 is a given initial point. Choosing an appropriate direction and a suitable step size are two basic steps of these algorithms. Generally, the search direction d_k is required to satisfy the descent condition $\nabla f(x_k)^T d_k < 0$ and α_k is determined such that it guarantees a sufficient reduction in function value. There are many different procedures to choose the search direction d_k. For example, Newton, quasi-Newton, conjugate gradient, steepest descent and trust region methods, see [20]. Among these methods, the Newton method has the highest rate of convergence where its direction is computed by solving system $G_k d_k = -g_k$ where $G_k = \nabla^2 f(x_k)$ and $g_k = \nabla f(x_k)$.

Computation of G_k or G_k^{-1}, in each iteration, is expensive or even could be analytically unavailable. Quasi-Newton methods were proposed to overcome this drawback without explicitly evaluating the Hessian. In these methods, B_k is an approximation to the Hessian that is updated at every iteration by means

2010 *Mathematics Subject Classification*. 90C30; 90C53
Keywords. Unconstrained optimization, BFGS method, Quasi-Newton method, Nonmonotone line search, Global convergence
Received: 04 April 2014; Accepted: 08 August 2014
Communicated by Predrag Stanimirović
Email addresses: kamini@razi.ac.ir (Keyvan Amini), bahrami.somaye@gmail.com (Somayeh Bahrami), shadi_24_63@yahoo.com (Shadi Amiri)

of a low-rank formula based only on the function and gradient values gathered during the descent process. The standard quasi-Newton methods generally meet the following secant equation:

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

where $s_k = x_{k+1} - x_k$, $y_k = g_{k+1} - g_k$ and B_{k+1} is an approximation of G_k which at the first iterate, B_0 is an arbitrary nonsingular positive definite matrix. Nowadays, Among quasi-Newton methods, the most efficient quasi-Newton method is perhaps the BFGS method which was proposed by Broyden, Fletcher, Goldfarb and Shanno independently. The matrix B_{k+1} in the BFGS method can be updated by the following formula:

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

It is known that the BFGS method preserves the positive definiteness of the matrices $\{B_k\}$, if the curvature condition $s_k^T y_k > 0$ holds. Therefore the BFGS direction is the descent direction of f at x_k no matter whether G_k is positive definite or not.

The convergence properties of the BFGS method for convex minimization have been well studied, for instance see [3, 4, 21]. Dai constructed an example with six cycling points and showed by it that the BFGS method with the Wolfe line search may fail for nonconvex functions[5]. Later, Mascarenhas presented a three-dimensional counter-example such that the BFGS method does not converge even with exact line search [17]. We note that many of studies have been focused on convex objective functions. To improve the global convergence property of the BFGS method, many modifications have been proposed, for instances Li and Fukushima made some modifications on the standard BFGS method and introduced a modified BFGS algorithm (MBFGS) [13, 14]. Under appropriate conditions, the globally and superlinearly convergence of their method have been proved for nonconvex optimization problems. Their modifications were so useful that have motivated many researchers to make further improvements on the BFGS method. For example, Xiao et al. introduced a new algorithm by using the MBFGS update formula suggested by Li and Fukushima along with a nonmonotone line search proposed in [23]. They proved that the method is globally convergent for nonconvex optimization problems.

As mentioned, another factor making a good iterative process is an appropriate line search which produces a sufficient reduction in function value. There are many conditions namely, Armijo, Wolfe or Goldstein condition. Among these conditions, Armijo rule is the most popular condition to accept a steplength stating as follows:

$$f(x_k + \alpha_k d_k) \le f_k + \sigma \alpha_k g_k^T d_k, \tag{3}$$

in which $\sigma \in (0, 1)$ and α_k is the largest member in $\{1, \rho, \rho^2, \cdots\}$ satisfying (3) such that $\rho \in (0, 1)$. In the mentioned formula f_k denotes $f(x_k)$ and it is clear that $f_{k+1} < f_k$ for every descent directions, so this schema is called a monotone line search.

The first nonmonotone line search technique was proposed by Grippo et al. for Newton's method by relaxing Armijo condition [10]. It was defined as follows:

$$f(x_k + \alpha_k d_k) \le \max_{0 \le j \le m(k)} \{f_{k-j}\} + \sigma \alpha_k g_k^T d_k, \tag{4}$$

where $0 \le m(k) \le \min\{m(k-1)+1, N\}$ that N is a nonnegative integer constant. In fact, in nonmonotone line search procedures some growth in the function value is permitted. As pointed out by many researchers, for example [6, 10, 11, 18, 22, 25], nonmonotone schemas not only can enhance the likelihood of finding a global optimum but also can improve speed of convergence in cases where a monotone schema is forced to creep along the bottom of a narrow curved valley. Although the nonmonotone techniques based on (4) have some advantages and work well in many cases, they include some drawbacks, see [6, 25]. One of the efficient nonmonotone line search methods to overcome these drawbacks has been proposed by Zhang and Hager in [25]. It has the same general schema as Grippo et al. while the statement "max" is replaced with a weighted average of function values over successive iterations. In detail, their nonmonotone line search is described as follows:

$$f(x_k + \alpha_k d_k) \leq C_k + \sigma\alpha_k g_k^T d_k, \tag{5}$$

where

$$\begin{aligned} C_k &= \begin{cases} f_k, & \text{if } k = 0 \\ (\eta_{k-1}Q_{k-1}C_{k-1} + f_k)/Q_k, & \text{if } k \geq 1 \end{cases} \\ Q_k &= \begin{cases} 1, & \text{if } k = 0 \\ \eta_{k-1}Q_{k-1} + 1, & \text{if } k \geq 1, \end{cases} \end{aligned} \tag{6}$$

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 results have been showed that the nonmonotone line search (5) is more efficient than the line search of Grippo et al. [25].

The nonmonotone BFGS method was first studied by Liu, et al. in [15]. Subsequently, two other nonmonotone BFGS methods were proposed for solving problem (1) in [12, 16]. Note that convergence analysis in all these algorithms was proved under convex assumption on the objective function. In this paper, a nonmonotone MBFGS algorithm is introduced and the global convergence of the method is proved without convexity assumption. Actually, the algorithm combines the MBFGS method, proposed by Xiao et al. in [23], with nonmonotone line search (5) and also gains advantages of [2] and [24]. Numerical experiments indicate that the new algorithm is promising and efficient.

This paper is organized as follows. The new algorithm is described in section 2. The convergence properties of the algorithm is proved in Section 3. Section 4 is dedicated to the numerical experiments. Finally, some conclusions are delivered in the last section.

2. The Nonmonotone Modified BFGS Algorithm

Although the BFGS algorithm is one of the most successful algorithms for unconstrained nonlinear optimization, it is well known that this method has two important disadvantages. First, the BFGS directions may not be descent especially when the condition $s_k^T y_k > 0$ isn't satisfied and so can not guarantee positive definiteness of the matrix B_k. Second, Although global and superlinear convergence results have been established for convex problems, it has been proved that, for general problems, the BFGS algorithm may not be convergent for nonconvex objective functions.

In this section, a nonmonotone MBFGS algorithm for nonconvex objective functions is presented guaranteeing the positive definiteness of the matrix B_k. The new method is introduced after describing some motivations.

As mentioned in the previous section, Li and Fukushima, in [13], introduced the modified secant equation

$$B_{k+1}s_k = y_k^*, \tag{7}$$

where

$$y_k^* \triangleq y_k + t_k^* s_k, \tag{8}$$

and

$$t_k^* = \overline{C}||g_k||^\mu + \max\{\frac{-s_k^T y_k}{||s_k||^2}, 0\} \geq 0, \tag{9}$$

where $\overline{C}$ and μ are two positive constants. Based on (7), they reformed the BFGS update formula as follows:

$$B_{k+1} = B_k - \frac{B_k s_k s_k^T B_k}{s_k^T B_k s_k} + \frac{y_k^* y_k^{*T}}{s_k^T y_k^*},$$

and introduced an efficient algorithm that is called MBFGS. It is easily seen that

$$s_k^T y_k^* \geq \overline{C}\|g_k\|^{\mu}\|s_k\|^2 > 0, \tag{10}$$

for all $k \in \mathbb{N}$. This property is independent on the convexity of f as well as the used line search and guarantees positive definiteness of the matrix B_k, see [23]. Following that, Xiao et al. combined the MBFGS algorithm with the nonmonotone line search (4) and constructed another MBFGS algorithm,[23]. They proved that this MBFGS methods possess a global convergence property even without convexity assumption on the objective function.

Under other circumstances, Yuan, in [24], proposed another modified BFGS algorithm for unconstrained optimization in which B_k is updated by the relation

$$B_{k+1} = B_k - \frac{B_k s_k s_k^T B_k}{s_k^T B_k s_k} + \tilde{t}_k \frac{y_k (y_k)^T}{s_k^T y_k}, \tag{11}$$

where

$$\tilde{t}_k = \frac{2}{s_k^T y_k}(f_k - f_{k+1} + s_k^T g_{k+1}). \tag{12}$$

The algorithm preserves the global and local superlinear convergence properties for convex objective functions, too.

Now, a new algorithm is going to be proposed in which an update formula for the BFGS method using y_k^* in equation (8) is presented then similar to (11), a parameter $\tau > 0$ is embeded into the update formula for computing B_k as follows:

$$B_{k+1} = B_k - \frac{B_k s_k s_k^T B_k}{s_k^T B_k s_k} + \tau \frac{y_k^* (y_k^*)^T}{s_k^T y_k^*}. \tag{13}$$

The mentioned B_k satisfies the following modified secant condition

$$B_{k+1} s_k = \tau y_k^*. \tag{14}$$

Remark 2.1. *Obviously, if $\tau = 1$, (14) reduces to the modified secant condition (7). Furthermore, a suitable choice for τ is $\tilde{t}_k$ in (12).*

Remark 2.2. *Since $\tau > 0$ and the inequality (10) holds, it is concluded that B_{k+1} generated by (13) is a positive definite matrix when B_k is a positive definite matrix.*

We now outline the new nonmonotone MBFGS algorithm as follows:

Algorithm N-MBFGS: (Nonmonotone Modified BFGS algorithm)

Input:. An initial point $x_0 \in \mathbf{R}^n$, a symmetric positive definite matrix $B_0 \in \mathbf{R}^{n\times n}$, constants $\sigma, \rho \in (0, 1)$ and the positive constants $\overline{C}$, μ and ϵ.

Step 0. Set $Q_0 = 1$, $C_0 = f_0$ and $k = 0$.

Step 1. If $\|g_k\| \leq \epsilon$, stop.

Step 2. Compute direction d_k by solving $B_k d_k = -g_k$.

Step 3. Set $\alpha_k = \rho^{j_k}$ where j_k is the smallest non-negative integer such that α_k satisfies (5).

Step 4. Set $x_{k+1} = x_k + \alpha_k d_k$.

Step 5. Select an appropriate τ. Update B_k by (13) in which y_k^* is obtained by (8) .

Step 6. Set $k = k + 1$ and go to Step 1.

3. Convergence Analysis

In this section, we discuss the global convergence properties of the new algorithm for general nonlinear objective function. We need to make the following assumptions on the objective function f.

(H1) The level set $L(x_0) = \{x \in \mathbb{R}^n | f(x) \leq f(x_0)\}$ is bounded.
(H2) $f(x)$ is differentiable on some neighborhood L_0 of $L(x_0)$ and its gradient g is Lipschitz continuous on L_0, namely, there exists a constant $L > 0$ such that

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

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

Lemma 3.1. *Suppose Assumption H1 is satisfied and the sequence $\{x_k\}$ is generated by Algorithm N-MBFGS, then $f_k \leq C_k$, for each $k \in \mathbb{N} \cup \{0\}$. Also the nonmonotone line search (5) is well-defined.*

Proof. Because $f(x)$ is a continuous function, by considering Assumption H1, it is deduced that $f(x)$ is bounded. Then the proof is followed Similar to Lemma 1.1 in [25].

Lemma 3.2. *Suppose the sequence $\{x_k\}$ is generated by Algorithm N-MBFGS, then $\{C_k\}$ is a nonincreasing sequence and for all $k \in \mathbb{N} \cup \{0\}$*

$$\{x_k\} \subset L(x_0). \tag{15}$$

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, according to Theorem 3.1 in [25], $\{C_k\}$ is a non-increasing sequence and

$$f_{k+1} \leq C_k \leq C_{k-1} \leq \dots \leq C_0 = f_0.$$

This implies that $\{x_k\}$ generated by Algorithm N-MBFGS is contained in the level set $L(x_0)$.

Lemma 3.3. *Let Assumptions H1 and H2 are satisfied and the sequence $\{x_k\}$ is generated by Algorithm N-MBFGS. If $\|g_k\| \geq \zeta$ holds for all $k \in \mathbb{N}$ with a constant $\zeta > 0$, then there exist positive constants β_1, β_2 and β_3 such that, for all $k \in \mathbb{N}$, the inequalities*

$$\|B_i s_i\| \leq \beta_1 \|s_i\| \ \ , \ \ \beta_2 \|s_i\|^2 \leq s_i^T B_i s_i \leq \beta_3 \|s_i\|^2, \tag{16}$$

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

Proof. We first show that there exist two positive constants m and M such that

$$\frac{y_k^{*^T} s_k}{\|s_k\|^2} \geq m, \tag{17}$$

and

$$\frac{\|y_k^*\|^2}{y_k^{*^T} s_k} \leq M. \tag{18}$$

To do this, from (10) and assumption $\|g_k\| \geq \zeta$, we have

$$s_k^T y_k^* \geq \bar{C}\|g_k\|^\mu \|s_k\|^2 \geq \bar{C}\zeta^\mu \|s_k\|^2, \tag{19}$$

so

$$\frac{y_k^{*^T} s_k}{\|s_k\|^2} \geq m,$$

where $m := \bar{C}\zeta^{\mu}$ is a positive constant.
Besides, it follows from (8), (9) and Cauchy-Schwartz inequality that

$$\|y_k^*\| \leq \|y_k\| + \|s_k\|(\bar{C}\|g_k\|^{\mu} + \frac{\|y_k\|}{\|s_k\|}).$$

Considering the relation (15) and Assumptions H1 and H2, there exists a constant $\bar{M} > 0$ such that $\|g_k\| \leq \bar{M}$. Therefore, it can be seen that

$$\|y_k^*\| \leq \|s_k\|(L + \bar{C}\bar{M}^{\mu} + L) = c\|s_k\|, \tag{20}$$

where L is Lipschitz constant in Assumption H2 and $c = L + \bar{C}D^{\mu} + L$. The relation (19) along with (20), for all $k \in \mathbb{N}$, result

$$\frac{\|y_k^*\|^2}{y_k^{*^T} s_k} \leq M,$$

where $M = \frac{c^2}{\bar{C}\zeta^{\mu}}$. The rest of the proof follows from (17), (18) and Theorem 2.1 in [3].

Lemma 3.4. *Let Assumptions H1 and H2 are satisfied and the sequence $\{x_k\}$ is generated by Algorithm N-MBFGS. If $\|g_k\| \geq \zeta$ holds for all $k \in \mathbb{N}$ with some constant $\zeta > 0$, then there is a positive constant $\overline{\alpha}$ such that $\alpha_k > \overline{\alpha}$ for all k belonging to $J = \{k \in \mathbb{N}|$ (16) holds$\}$.*

Proof. It is sufficient that the case $\alpha_k \neq 1$ is considered. The line search rule (5) implies that $\acute{\alpha}_k \equiv \alpha_k/\rho$ does not satisfy inequality (5), i.e.

$$f(x_k + \acute{\alpha}_k d_k) - C_k > \sigma \acute{\alpha}_k g_k^T d_k,$$

because $f_k \leq C_k$, it is concluded that

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

Using the mean-value theorem, it is obtained

$$g(x_k + \theta \acute{\alpha}_k d_k)^T d_k > \sigma g_k^T d_k, \tag{21}$$

where $\theta \in (0, 1)$. Now, According to the Cauchy-Schwartz inequality, Assumption H2, (16) and (21), it follows that

$$\begin{aligned}
\acute{\alpha}_k L\|d_k\|^2 &\geq \|g(x_k + \theta\acute{\alpha}_k d_k) - g_k\|.\|d_k\| \\
&\geq (g(x_k + \theta\acute{\alpha}_k d_k) - g_k)^T d_k \\
&> -(1 - \sigma) g_k^T d_k \\
&= (1 - \sigma) d_k^T B_k d_k \\
&\geq (1 - \sigma)\beta_2 \|d_k\|^2.
\end{aligned}$$

So $\acute{\alpha}_k > \beta_2(1 - \sigma)/L$. This means that $\alpha_k > \overline{\alpha}$, for all $k \in J$, where $\overline{\alpha} = \beta_2\rho(1 - \sigma)/L$ is positive.

Lemma 3.5. *Suppose that Assumption H1 is satisfied and the sequence $\{x_k\}$ is generated by Algorithm N-MBFGS, then*

$$\sum_{k=0}^{\infty}(-g_k^T s_k) < \infty. \tag{22}$$

Proof. Using (5) and (6), it can be concluded that

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

where the last equality is ensued from $Q_{k+1} = \eta_k Q_k + 1$. This means that

$$C_k - C_{k+1} \geq \frac{-\sigma\alpha_k g_k^T d_k}{Q_{k+1}}. \tag{23}$$

On the other hand, it was proved in [25] that

$$Q_{k+1} = 1 + \sum_{j=0}^{k}\prod_{i=0}^{j}\eta_{k-i} \leq 1 + \sum_{j=0}^{k}\eta_{\max}^{j+1} \leq \sum_{j=0}^{\infty}\eta_{\max}^{j} = \frac{1}{1-\eta_{\max}}. \tag{24}$$

Inequalities (23) and (24) imply

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

therefore

$$\sum_{k=0}^{\infty}(C_k - C_{k+1}) \geq \sigma(1-\eta_{\max})\sum_{k=0}^{\infty}(-\alpha_k g_k^T d_k). \tag{25}$$

This inequality along with Lemma 3.1 indicate

$$\sigma(1-\eta_{\max})\sum_{k=0}^{\infty}(-\alpha_k g_k^T d_k) \leq C_0 - \lim_{k\to\infty} C_k \leq f_0 - \lim_{k\to\infty} f_k < \infty,$$

where the last inequality comes from Assumption H1 and this fact that $f(x)$ is a continuous function. So (22) holds and the proof is completed.

Now, the main result of this section, the global convergence of the new algorithm, can be described.

Theorem 3.6. *Suppose Assumptions H1 and H2 are satisfied and the sequence $\{x_k\}$ is generated by Algorithm N-MBFGS, then*

$$\liminf_{k\to\infty} \|g_k\| = 0. \tag{26}$$

Proof. CONTRADICTION. Assume that $\liminf_{k\to\infty}\|g_k\| \neq 0$, so there exists a constant $\zeta > 0$ such that

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

for all k sufficiently large. Since $B_k s_k = \alpha_k B_k d_k = -\alpha_k g_k$, it follows from (22) that

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

$\|g_k\| \geq \zeta$ and considering the definition of J in Lemma 3.4, lead us to

$$\begin{aligned}\sum_{k=0}^{\infty} \alpha_k \frac{s_k^T B_k s_k}{\|B_k s_k\|^2} \|g_k\|^2 &\geq \zeta^2 \sum_{k=0}^{\infty} \alpha_k \frac{s_k^T B_k s_k}{\|B_k s_k\|^2} \\ &\geq \zeta^2 \sum_{k \in J} \alpha_k \frac{s_k^T B_k s_k}{\|B_k s_k\|^2} \\ &> \zeta^2 \overline{\alpha} \sum_{k \in J} \frac{s_k^T B_k s_k}{\|B_k s_k\|^2},\end{aligned}$$

in which the last inequality comes from Lemma 3.4. This implies

$$\sum_{k \in J} \frac{s_k^T B_k s_k}{\|B_k s_k\|^2} < \infty. \tag{27}$$

Since the set J is infinite, it is concluded that $\frac{s_k^T B_k s_k}{\|B_k s_k\|^2} \to 0$ 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 (16).

4. Numerical Experiments

In this section, the numerical experiments of the new MBFGS algorithm (N-MBFGS) are compared with the standard BFGS algorithm along with Armijo line search (BFGS) and the MBFGS algorithm proposed by Xiao et al., (MBFGS-XG)[23]. To have an appropriate comparison, we introduce another algorithm in which the nonmonotone line search proposed by Zhang and Hager is replaced by Xiao's nonmonotone line search and named it MBFGS-XZH.

The experiments are written in MATLAB R2009a programming environment with double precision format. We tested all the algorithms on a set of 83 problems that were taken of Andrei [1] and Moré, et al. [19]. For a better comparison, the same stopping criterion and the same parameters were used in all the algorithms . The stopping criterion is

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

and as far as our experiences show, the parameters $\sigma = 0.38$, $\rho = 0.46$, $\eta_k = 0.2$ and $\tau_k = 0.1$ have the best results for all the algorithms. In addition, similar to [23], $N = 5$, $\mu = 4$ and $\bar{C} = 10^{-2}$ if $\|g_k\| \leq 10^{-2}$ and $\bar{C} = 0$ otherwise, are chosen.

Tables 1 and 2 demonstrate the numerical results of the algorithms in which 'Prob. name' and 'Dim' present the name and the dimension of the test problems, respectively. Furthermore, the symbols N_i and N_f in Table 1 stand for the number of iterations and the number of function evaluations, respectively. Also, Table 2 presents the required CPU time of the algorithms. Clearly, in the considered algorithms, the number of iterates and the number of gradient evaluations are equal. Therefore, the number of gradient evaluations is not included in the tables. Furthermore, the symbol "NaN" in the tables means that the direction d_k could not be computed by $d_k = -B_k^{-1} g_k$.

Table 1. Numerical results of the N_i and N_f

Prob. name	Dim	BFGS N_i/N_f	MBFGS-XG N_i/N_f	MBFGS-XZH N_i/N_f	N-MBFGS N_i/N_f
1. Powell b. scaled	2	111/185	148/210	131/ 197	111/445
2. Brown b. scaled	2	12/51	11/ 48	11/ 49	38/137
3. Beale	2	14 / 25	14 / 22	14 / 23	20/ 65
4. Helical valley	3	25/57	28/ 52	27/ 52	27/106
5. Gaussian	3	4 /8	4/ 8	4/8	19/ 54
6. Box 3-dim	3	NaN	37/ 57	39/ 61	28/97
7. Gulf research	3	22 / 39	NaN	24 / 34	29 / 86
8. Brown and...	4	18/72	20/ 70	19/69	19/ 100
9. Wood	4	77 / 144	91 / 136	20 / 51	71/ 282
10. Biggs EXP6	6	38/51	36/42	34/ 43	NaN
11. Watson	20	43/80	45/ 76	34/ 69	45/ 159
12. VARDIM	50	16 /51	16/ 51	16 / 51	8/ 51
13. Variably dim...	100	16 / 57	16/ 57	16 / 57	9/ 58
14. Penalty II	100	103 / 1196	128 / 1569	119 / 1357	115/ 1331
15. LIARWHD	100	23 / 79	178 / 429	20 / 59	18 / 63
15. LIARWHD	1000	28 / 105	245 / 821	37 / 126	19/ 73
16. Trigonometric	200	60/ 66	58 / 60	58/ 60	126/ 361
17. Raydan 1	200	68 / 251	200 / 500	66 / 211	60 / 196
17. Raydan 1	1000	136 / 761	475 / 2277	131/ 675	116 / 592
18. Ge. trid. 2	200	91 / 607	370 / 1383	92 / 596	46 / 318
19. Penalty I	500	21 / 184	149 / 892	48 / 302	19 / 149
19. Penalty I	1000	16 / 190	225/ 1809	52/ 380	22 / 202
20. Ex.q.p. QP1	900	21 / 52	25 / 45	22 / 53	22 / 77
20. Ex.q.p. QP1	1000	26 / 60	34 / 54	22 / 48	19/ 62
21. Ex.q.p. QP2	900	80 / 570	NaN	NaN	27 /104
21. Ex.q.p. QP2	1000	130 / 780	NaN	NaN	22/ 83
22. Ex. White...	980	1419 / 9238	3242/ 11967	1523/ 9027	1201/ 8470
23. DIXMAANA	999	7 / 11	8/10	8 / 10	19/ 48
24. Ex. Rosen.	1000	1085 / 6155	1734 / 7736	1079 / 5812	541 / 3519
25. Ex. Freu...	1000	18 / 94	288 / 1572	28 / 126	21/ 114
26. Ge. Rose.	1000	4232 /13481	NaN	1215/9677	461/4094
27. Ge. White...	1000	6689/ 17949	NaN	6823/16358	6607/ 32783
28. Ex. Beale	1000	16 / 40	51 / 83	24 / 48	21/ 67
29. Ex. PEN.	1000	16 / 186	199 / 1530	47 / 386	26 / 228
30. Per. quad.	1000	224 / 2023	1002/ 8203	224 / 2020	131 / 1183
31. Raydan 2	1000	5 / 7	5 / 7	5 / 7	16/ 43
32. Diagonal 1	1000	NaN	1359 / 8777	604 / 4947	171 / 1419
33. Diagonal 2	1000	194 / 195	194 / 195	194 / 195	209/ 526
34. Diagonal 3	1000	198 / 1610	1051 / 7640	199 / 1596	118 / 948
35. Diagonal 4	1000	2/9	2/9	2/9	18/ 57
36. Diagonal 5	1000	5 / 7	5 / 6	5 / 6	16/ 43
37. Diagonal 7	1000	5 / 8	6 / 8	6 / 8	17/ 46
38. Diagonal 8	1000	4 / 7	4 / 7	4 / 7	15/ 39
39. Diagonal 9	1000	637 / 5326	1309 / 8572	449/ 3645	142 / 1157
40. Hager	1000	35 / 145	326/ 1063	48 / 189	31 / 124
41. Ge. trid. 1	1000	52 / 242	290 / 1085	62 / 269	27 / 119
42. Ex. trid. 1	1000	20 / 27	19/ 24	20 / 26	24/ 70
43. Ex. trid. 2	1000	28 / 65	70 / 74	30 / 59	40/ 115
44. Ex. TET	1000	7 / 14	7 / 13	7 / 14	17/ 49
45. Ex. Him.	1000	13 / 41	168/ 589	23/ 59	18/ 60
46. Ge. PSC1	1000	52 / 119	260 / 574	95 /168	42/ 128
47. Ex. PSC1	1000	12 / 23	14 / 21	13/ 21	17/ 47
48. Ex. Powell	1000	82 / 342	802 / 3170	78 / 307	37 / 152
49. Full H. FH3	1000	3 / 14	3 / 14	3 / 14	15/ 49
50. Ex. BD1	1000	10/ 19	18 /21	12 /17	20/ 56
51. Ex. Maratos	1000	1555 / 8285	NaN	NaN	1006 / 6339
52. per.q.diag.	1000	15/ 63	123/ 308	27/ 86	17/ 54
53. Ex. Wood	1000	1592/ 10206	2444/ 10342	1602/ 10008	1310/ 10152
54. Quad. QF1	1000	200 / 1617	854 / 6390	201 / 1612	135 / 1084

Table 1. (*continued*)

55. Quad. QF2	1000	347 / 3481	1140 / 9918	365 / 3653	168/ 1683
56. Ex.q.ex.EP1	1000	3 / 12	3 / 12	3 / 12	17/ 66
57. FLETCHCR	1000	1502 /10249	1796 / 9995	1538 /10106	1237 / 10302
58. BDQRTIC	1000	112 / 683	776 / 3917	137 / 785	69 / 424
59. TRIDIA	1000	528 / 6228	1062 / 10652	503 / 5841	374/ 4419
60. ARWHEAD	1000	7 / 22	7/ 20	7 / 21	11/ 44
61. NONDIA	1000	9/46	108/511	26/104	25/129
62. NONDQUAR	1000	39/123	184/ 726	45/ 139	17/ 59
63. DQDRTIC	1000	12 / 52	13 / 27	12 / 40	19/ 83
64. EG2	1000	58 / 82	59/ 75	61 / 80	63/ 229
66. Par. per. quad.	1000	150 / 1361	1022 / 8331	153 /1373	78 / 703
66. Al. per. quad.	1000	224 / 2023	1002/ 8206	225/ 2029	131/ 1183
67. Pert. trid. Q...	1000	214 / 1934	1043/ 8493	214/ 1930	133/ 1201
68. Stair. 1	1000	80/ 604	NaN	79 / 582	46/ 403
69. Stair. 2	1000	80 / 604	NaN	79 / 582	46/403
70. POWER	1000	1003/ 17351	1007/ 16550	1003 / 17244	1004/ 17259
71. ENGVAL1	1000	43 / 159	526 / 1400	76 / 245	23 / 84
72. EDENSCH	1000	48/ 191	428 / 1253	56 / 195	19/ 67
73. CUBE	1000	287 / 1244	NaN	216 / 1063	233 / 1246
74. BDEXP	1000	17 / 18	17 / 18	17 / 18	7/ 8
75. QUARTC	1000	16 / 20	16/20	16 / 20	6/ 14
76. DIXON3DQ	1000	518/1034	873/ 1156	524/ 1035	686/ 1783
77. Ex. DEN. B	1000	7 / 11	7 / 10	7 / 10	16/ 46
78. Ex. DEN. F	1000	13 / 67	292 / 1516	30 / 141	26 / 131
79. BIGGSB1	1000	518 / 1035	883 / 1167	521 / 1024	689 / 1794
80. Ge. Quad.	1000	NaN	NaN	NaN	19/ 57
81. SINCOS	1000	12 / 23	14 / 21	13 / 21	17/ 47
82. HIMMELBG	1000	20 / 24	18/ 21	20/ 24	7/ 14
83. HIMMELH	1000	6 / 11	6 / 10	6 / 11	17/ 49

Table 2.Numerical results of CPU time

Prob. name	Dim	BFGS	MBFGS-XG	MBFGS-XZH	N-MBFGS
1. Powell b. scaled	2	0.04680029	0.04680030	0.03120020	0.07800050
2. Brown b. scaled	2	0.01560010	0	0	0.01560010
3. Beale	2	0	0.01560010	0	0.01560010
4. Helical valley	3	0	0	0	0
5. Gaussian	3	0	0	0	0
6. Box 3-dim	3	NaN	0.01560010	0.01560010	0.01560010
7. Gulf research	3	0.01560010	NaN	0.03120020	0.01560010
8. Brown and...	4	0.01560010	0.01560010	0.01560010	0
9. Wood	4	0.01560010	0.01560010	0.03120020	0.01560010
10. Biggs EXP6	6	0.01560010	0.01560009	0	NaN
11. Watson	20	0.04680030	0.04680030	0.03120020	0.07800050
12. VARDIM	50	0	0.01560010	0.01560010	0
13. Variably dim...	100	0.04680030	0.06240040	0.03120020	0.06240040
14. Penalty II	100	0.98280629	1.18560759	0.98280629	1.10760709
15. LIARWHD	100	0.09360060	0.37440239	0.06240040	0.07800050
15. LIARWHD	1000	7.95605099	70.71525330	10.43646690	5.55363560
16. Trigonometric	200	1.41960910	1.38840890	1.35720870	3.18242040
17. Raydan 1	200	0.46800299	1.57561010	0.53040339	0.54600349
17. Raydan 1	1000	0.39218651e+02	1.37936084e+02	0.38017443e+02	0.35459027e+02
18. Ge. trid. 2	200	0.63960409	2.63641689	0.68640440	0.37440240
19. Penalty I	500	2.88601849	17.62811299	5.61603599	2.19961410
19. Penalty I	1000	0.08065251e+02	1.04848272e+02	0.24351756e+02	0.11450473e+02
20. Ex.q.p. QP1	900	4.83603099	5.99043840	5.03883229	5.21043339
20. Ex.q.p. QP1	1000	7.80004999	9.65646190	6.16203949	5.66283629
21. Ex.q.p. QP2	900	19.45332469	NaN	NaN	6.41164109
21. Ex.q.p. QP2	1000	37.93944319	NaN	NaN	6.56764209
22. Ex. White...	980	4.19174687e+02	9.86363122e+02	4.58455738e+02	3.72280786
23. DIXMAANA	999	2.13721370	2.23081430	2.29321469	5.94363809
24. Ex. Rosen.	1000	4.33479978e+02	6.71896307e+02	4.35897994e+02	2.22691427e+02

Table 2. (*continued*)

25. Ex. Freu...	1000	5.13243289	85.87855049	7.97165109	6.11523920
26. Ge. Rose.	1000	1.25105001e+03	NaN	0.36764755e+03	0.14387972e+03
27. Ge. White...	1000	1.93433439e+03	NaN	2.01275610e+03	2.03960387e+03
28. Ex. Beale	1000	4.92963159	15.11649689	6.84844389	6.08403900
29. Ex. PEN.	1000	4.53962909	58.23517330	13.54088679	7.61284879
30. Per. quad.	1000	0.68156836e+02	3.17072032e+02	0.69810447e+02	0.41543066e+02
31. Raydan 2	1000	1.20120769	1.43520919	1.27920820	4.61762960
32. Diagonal 1	1000	NaN	3.98192552e+02	1.78387143e+02	0.51277528e+02
33. Diagonal 2	1000	55.84835799	55.72355719	55.52075590	62.4940005
34. Diagonal 3	1000	0.57501968	3.130940069	0.57860770	0.36098631
35. Diagonal 4	1000	0.46800299	0.436802799	0.43680279	5.42883480
36. Diagonal 5	1000	1.26360810	1.404009000	1.38840889	4.92963159
37. Diagonal 7	1000	1.35720870	1.71601099	1.77841140	5.05443240
38. Diagonal 8	1000	0.98280629	1.2168078	1.24800799	4.64882979
39. Diagonal 9	1000	1.93831242	3.90891705	1.33942458	0.43945481
40. Hager	1000	10.24926570	93.9438021	14.18049090	9.09485820
41. Ge. trid. 1	1000	16.56730619	86.62735529	18.04931569	7.98725120
42. Ex. trid. 1	1000	5.818837299	5.44443490	5.78763710	7.05124520
43. Ex. trid. 2	1000	7.924850799	20.20212950	8.56445489	11.99647689
44. Ex. TET	1000	1.918812299	1.95001249	1.85641190	5.03883229
45. Ex. Him.	1000	3.978025500	49.73311880	6.63004250	5.47563509
46. Ge. PSC1	1000	15.303698100	78.92090590	28.15818050	13.07288388
47. Ex. PSC1	1000	3.416421899	4.05602599	3.60362309	5.02323219
48. Ex. Powell	1000	0.235405509e+02	2.35733111e+02	0.22464144e+02	0.10842069e+02
49. Full H. FH3	1000	0.717604599	0.68640440	0.73320469	4.88283130
50. Ex. BD1	1000	2.745617599	5.8032372	3.69722369	5.85003749
51. Ex. Maratos	1000	4.582997378e+02	NaN	NaN	3.04467151e+02
52. per.q.diag.	1000	4.352427899	35.67742870	7.45684779	4.77363060
53. Ex. Wood	1000	4.662245885e+02	7.21442224e+02	4.73073032e+02	3.92935318e+02
54. Quad. QF1	1000	0.585315752e+02	2.61722877e+02	0.58453574e+02	0.41184264e+02
55. Quad. QF2	1000	1.037250648e+02	3.49255038e+02	1.08342694e+02	0.50996726e+02
56. Ex.q.ex.EP1	1000	0.717604600	0.7956051000	0.67080429	4.94523170
57. FLETCHCR	1000	4.306407605e+02	5.14288496e+02	4.42793238e+02	3.70034372e+02
58. BDQRTIC	1000	0.316682030e+02	2.23643033e+02	0.39125050e+02	0.203737306e+02
59. TRIDIA	1000	1.531773819e+02	3.11315595e+02	1.44098123e+02	1.11540715e+02
60. ARWHEAD	1000	1.887612099	1.85641190	1.88761209	3.13562009
61. NONDIA	1000	2.480415900	32.01140519	7.39444739	7.50364809
62. NONDQUAR	1000	11.528473899	53.92954569	12.87008250	4.92963159
63. DQDRTIC	1000	3.354021499	3.58802299	3.27602100	5.89683780
64. EG2	1000	16.520505900	17.33171109	17.33171109	19.29732370
65. Par. per. quad.	1000	46.61309879	3.24996883e+02	46.65989910	25.61536420
66. Al. per. quad.	1000	0.67096030e+02	3.03109942e+02	0.66019623e+02	0.39967456e+02
67. Pert. trid. Q...	1000	0.64163211e+02	3.20691255e+02	0.63820009e+02	0.40575860e+02
68. Stair. 1	1000	24.97576010	NaN	24.30495579	27.92417899
69. Stair. 2	1000	24.80415900	NaN	24.44535670	29.70259039
70. POWER	1000	3.10972393e+02	3.13858411e+02	3.13702410e+02	3.24029677e+02
71. ENGVAL1	1000	0.12121277e+02	1.51383370e+02	0.21668538e+02	0.06661242e+02
72. EDENSCH	1000	0.14055690e+02	1.26594811e+02	0.16863708e+02	0.06988844e+02
73. CUBE	1000	0.84443341e+02	NaN	0.63960410e+02	0.70855654e+02
74. BDEXP	1000	4.80483079	4.63322969	4.71123019	1.88761212
75. QUARTC	1000	4.43042839	4.53962909	4.47722869	1.65361060
76. DIXON3DQ	1000	1.46625339e+02	2.55248836e+02	1.46422538e+02	1.98714073e+02
77. Ex. DEN. B	1000	1.87201199	1.85641190	1.93441239	4.80483079
78. Ex. DEN. F	1000	3.66602350	84.66174269	8.68925570	7.78444990
79. BIGGSB1	1000	1.48902954e+02	2.54640432e+02	1.45829734e+02	2.03862106e+02
80. Ge. Quad.	1000	NaN	NaN	NaN	5.49123520
81. SINCOS	1000	3.41642190	4.29002749	3.75962410	4.97643189
82. HIMMELBG	1000	5.69403649	4.96083180	5.75643689	1.99681280
83. HIMMELH	1000	1.63801050	1.60681030	1.60681030	5.07003250

To have a comprehensive comparison among the reported results of the tables, the proposed performance profiles from Dolan and Moré in [8] is exploited in the sense of the number of iterations, N_i, function

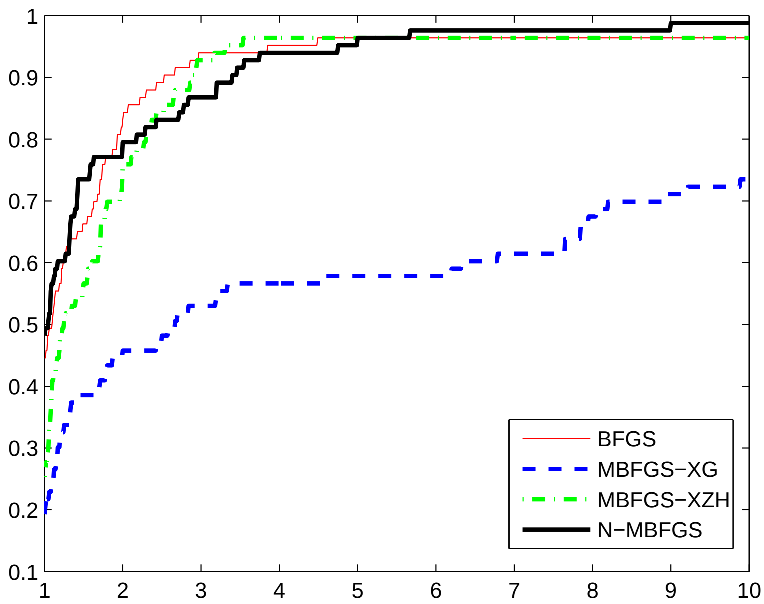

Figure 1: Total number of iterations performance profiles for the presented algorithms

evaluations, N_f, and CPU time.

The performance profile can be considered as a tool for evaluating and comparing the performances of iterative algorithms, where the profile of each code is measured based on the ratio of its computational outcome versus the computational outcome of the best presented code. It is known that a plot of the performance profile reveals all of the major performance characteristics which is a common tool to graphically compare the effectiveness as well as the robustness of the algorithms. One of the properties of this profile is the first point to the left side of the graphs that indicates the percentage of test problems for which a method is the fastest. The other property is the highest point in the right side of the graphs that shows the success rate of algorithms in solving problems . We also can see, an algorithm growing up more faster than other considered algorithms, it means in the cases that an algorithm is not the best algorithm which performance index is close to performance index of the best algorithm, please see [8] for more details.

Figures 1-3 obviously exhibit that N-MBFGS algorithm has a better performance than the MBFGS-XG algorithm proposed by Xiao et.al and it is competitive with the other algorithms.

Also, it is shown in Figure 1 that N-MBFGS algorithm has more than 48% of the minimum number of iterations to solve the problems when BFGS algorithm is about 44% and MBFGS-XG and MBFGS-XZH algorithms are only 19% and 25%, respectively. Secondly, the proposed algorithm solves problems more successfully than the others.

Although Figure 2 shows BFGS and MBFGS-XZH algorithms grow up faster than N-MBFGS algorithm, that means when these algorithms are not the best one their function evaluations are close to the best algorithm, it demonstrates that N-MBFGS algorithm has the best algorithm respecting to the minimum number of evaluation N_f , about 39% of the mentioned problems more than the others.

Obviously, N-MBFGS in Figure 3 has the best results regarding to the most wins when the performance measure is CPU time, approximately 42%. It means that N-MBFGS algorithm can solve 42% of the problems in the least time in comparison with the other. Also, considering the ability of completing the run successfully, it can be seen that N-MBFGS has better results in comparison with the others, however it is still in competition with BFGS and MBFGS-XZH.

Altogether, it is deduced that two algorithms we have presented, N-MBFGS and MBFGS-XZH, works

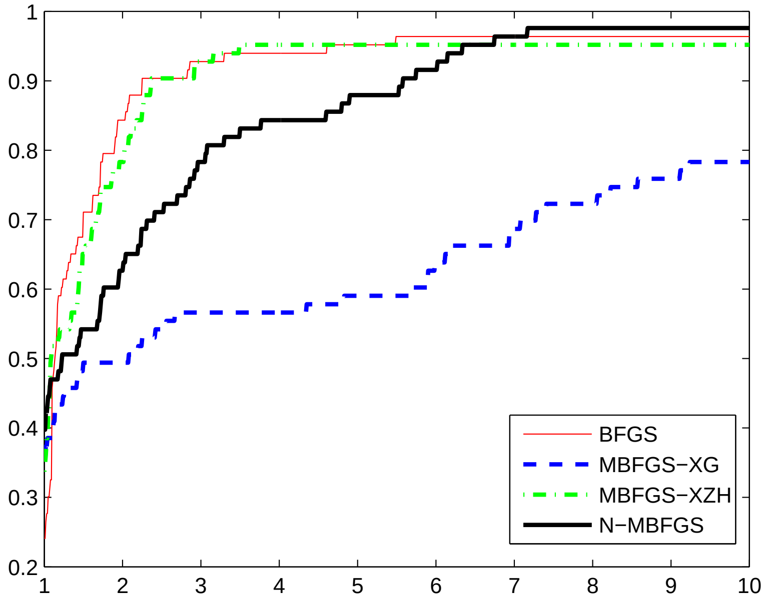

Figure 2: Total number of function evaluations performance profiles for the presented algorithms

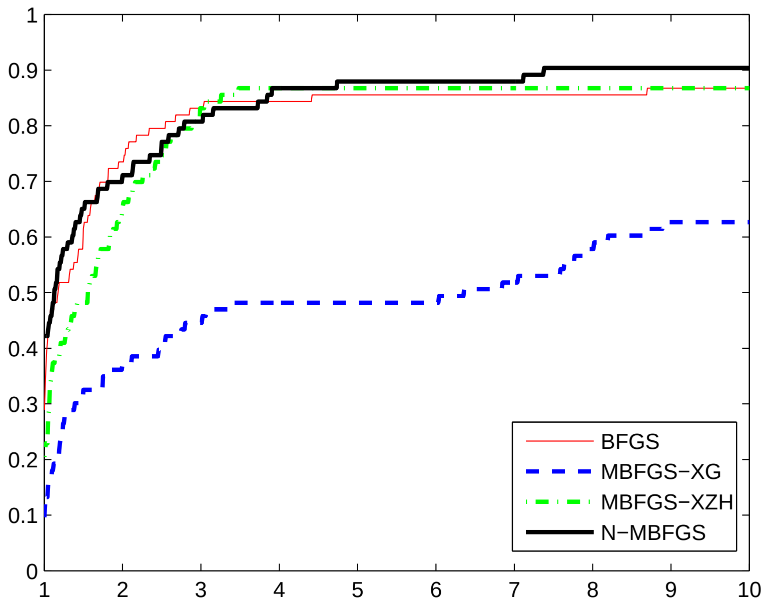

Figure 3: *CPUtime* performance profiles for the presented algorithms

as well as BFGS algorithm and it performs much better than MBFGS-XG algorithm.

5. Conclusions

In this paper, by proposing a modified BFGS update to approximate Hessian matrix and combining it with a known nonmonotone line search strategy, we have introduced a new nonmonotone BFGS algorithm for nonconvex unconstrained optimization problems. It is well-known that the nonmonotone schemas not only can improve the likelihood of finding a global optimum but also can enhance speed of convergence especially in presence of a narrow curved valley, so we are interested to getting benefit from their properties in our algorithm. Finally, the globally convergence of the algorithm is proved for nonconvex unconstrained problems and numerical results are presented to show that the proposed algorithm is competitive with the standard BFGS method and is more efficient than the nonmonotone BFGS algorithm proposed by Xiao et al. in [23].

References

[1] N. Andrei, An Unconstrained Optimization Test Functions Collection, Adv. Model. Optim. 10(1) (2008) 147–161.
[2] M. Al-Balli, H.Khalfan, A combined class of self–scaling and modified Quasi-Newton methods. Presented at the conference NAO2008 on Numerical Analysis and Optimization, Muscat, 6–8 April 2008.
[3] R.H. Byrd and J. Nocedal, A tool for the analysis of quasi-Newton methods with application to unconstrained minimization. SIAM J. Numer. Anal. 26 (1989) 727–739.
[4] R.H. Byrd, J. Nocedal and Y. Yuan, Global convergence of a class of quasi-Newton methods on convex problems. SIAM J. Numer. Anal. 24 (1987) 1171–1189.
[5] Y. H. Dai, Convergence properties of the BFGS algorithm. SIAM J. Optim. 13 (2002) 693–701.
[6] Y. H. Dai, On the nonmonotone line search. J. Optim. Theory Appl. 112(2) (2002) 315–330.
[7] Y. Dai and Q. Ni, Testing different conjugate gradient methods for large-scale unconstrained optimization. J. Comput. Math 21(3) (2003) 311–320.
[8] E. Dolan and J.J. Moré, Benchmarking optimization software with performance profiles. Math. Program. 91 (2002) 201–213.
[9] A.Griewank, On automatic differentiation. In: Math. Program.: Recent developments and applications (M. Iri and K.Tanabe (Eds)), Dordrecht: Kluwer Academic, (1989) 84–108.
[10] L. Grippo, F. Lampariello and S. Lucidi, A nonmonotone line search technique for Newtons method. SIAM J. Numer. Anal. 23 (1986) 707–716.
[11] C. Gu and D. Zhu, A nonmonotone line search filter method with reduced Hessian updating for nonlinear optimization, J. Syst. Sci. Complex. 26 (2013) 534–555.
[12] J. Han and G. Liu, Global convergence analysis of a new nonmonotone BFGS algorithm on convex objective function. Comput. Optim. Appl. 7 (1997) 277–289.
[13] D.H. Li and M. Fukushima, A modified BFGS method and its global convergence in nonconvex minimization. J. Comput. Appl. Math. 129 (2001) 15–35.
[14] D.H. Li and M. Fukushima, On the global convergence of the BFGS method for nonconvex unconstrained optimization problems. SIAM J. Optim. 11 (2001) 1054–1064.
[15] G. Liu, J. Han and D. Sun, Global convergence of the BFGS algorithm with nonmonotone line search. Optimization. 34 (1995) 147–159.
[16] L. Liu, S. Yao and 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.
[17] W.F. Mascarenhas, The BFGS method with exact line searches fails for non-convex objective functions. Math. Program. 99 (2004) 49–61.
[18] M. Miladinović, P. Stanimirović and S. Miljković, Scalar Correction Method for Solving Large Scale Unconstrained Minimization Problems. J.Optim. Theory. Appl. 151 (2011) 304–320.
[19] J.J. Moré, B.S. Garbow and K.E. Hillstrom, Testing unconstrained optimization software. ACM Trans. Math. Soft. 7(1) (1981) 17–41.
[20] J. Nocedal and J.S. Wright, Numerical Optimization. Springer, New York ,2006.
[21] M.J.D. Powell, Some global convergence properties of a variable metric algorithm for minimization without exact line searches, Non-linear Programming, SIAM-AMS Proceedings, Vol. IX. (R.W. Cottle, C.E. Lemke (Eds)), American Mathematical Society, Providence 9 (1976) 53–72.
[22] Ph.L. Toint, An assessment of nonmonotone linesearch technique for unconstrained optimization. SIAM J. Sci. Comput. 17 (1996) 725–739.
[23] Y.Xiao, H.Sun and Z.Wang, A globally convergent BFGS method with nonmonotone line search for non-convex minimization. J. Comput. Appl. Math. 230 (2009) 95–106.
[24] Y.Z. Yuan, A modified BFGS algorithm for unconstrained optimization. IMA J. Numer. Anal. 11 (1991) 325–332.
[25] H.C. Zhang and W.W. Hager, A nonmonotone line search technique and its application to unconstrained optimization. SIAM J. Optim. 14 (2004) 1043–1056.