Journal of Computational and Applied Mathematics 230 (2009) 95–106

Contents lists available at ScienceDirect

Journal of Computational and Applied Mathematics

journal homepage: www.elsevier.com/locate/cam

A globally convergent BFGS method with nonmonotone line search for non-convex minimization[☆]

Yunhai Xiao[a,b,*], Huijuan Sun[b], Zhiguo Wang[b]

[a] *Department of Mathematics, Nanjing University, Nanjing, 210093, PR China*

[b] *Institute of Applied Mathematics, College of Mathematics and Information Science, Henan University, Kaifeng, 475004, PR China*

ARTICLE INFO

Article history:
Received 24 January 2008
Received in revised form 27 October 2008

Keywords:
Non-convex minimization
Secant equation
BFGS method
Nonmonotone line search
Global convergence

ABSTRACT

In this paper, we propose a modified BFGS (Broyden–Fletcher–Goldfarb–Shanno) method with nonmonotone line search for unconstrained optimization. Under some mild conditions, we show that the method is globally convergent without a convexity assumption on the objective function. We also report some preliminary numerical results to show the efficiency of the proposed method.

1. Introduction

We are concerned with quasi-Newton methods for finding a local minimum of the unconstrained optimization problem

$$\min_{x\in\mathbf{R}^n} f(x). \tag{1.1}$$

We assume that $f : \mathbf{R}^n \to \mathbf{R}$ is continuously differentiable. We denote the gradient of f at x_k by $g(x_k)$. For the sake of simplicity, we will abbreviate $f(x_k)$ and $g(x_k)$ as f_k and g_k, respectively. A quasi-Newton method for solving (1.1) generates a sequence of iterates $\{x_k\}$ as

$$x_{k+1} = x_k + \alpha_k d_k, \quad k = 0, 1, \ldots, \tag{1.2}$$

where α_k is a step length, and d_k is a solution of the system of linear equations:

$$B_k d_k + g_k = 0, \tag{1.3}$$

where B_k is an approximation of the Hessian matrix at x_k. We pay particular attention to the BFGS method in which B_k is updated by

$$B_{k+1} = B_k - \frac{B_k s_k s_k^T B_k}{s_k^T B_k s_k} + \frac{y_k y_k^T}{s_k^T y_k}, \tag{1.4}$$

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

☆ This work was supported by Chinese NSF grant 10761001, 10771057, and Henan University Science Foundation grant 07YBZR002.

* Corresponding author at: Department of Mathematics, Nanjing University, Nanjing, 210093, PR China.
E-mail addresses: yhxiao@henu.edu.cn (Y. Xiao), happy_hj@126.com (H. Sun), wangzg@henu.edu.cn (Z. Wang).

0377-0427/$ – see front matter Crown Copyright © 2008 Published by Elsevier B.V. All rights reserved.
doi:10.1016/j.cam.2008.10.065

The convergence properties of the BFGS method for convex minimization have been well studied (e.g. [2,3,20]). Dai (see[6]) constructed an example to show that the standard BFGS method with Wolfe line search may fail to converge. On the other hand, Mascaren (see [17]) presented an example that shows the nonconvergence of the standard BFGS method with exact line search. To improve the global convergence property of BFGS method, Li and Fukushima (see[13, 14]) made some slight modifications to the standard BFGS method and developed a modified BFGS method and a cautious BFGS method. Under appropriate conditions, both methods are globally and superlinearly convergent for non-convex minimization problems.

The earliest nonmonotone line search technique was proposed by Grippo, Lampariello, and Lucidi (see [9]) in Newton's method. It has received much attention since then (e.g. [1,5,7,10,11,19,21–24,27]). The nonmonotone BFGS method was first studied by Liu, Han, and Sun (see [15]), under the assumption that the objective functions are uniformly convex. They established the global convergence of the method. Subsequently, Han and Liu introduced another nonmonotone BFGS algorithm on convex objective functions (see [12]). Their numerical experiments showed that this method was competitive to the standard BFGS algorithm. More recently, Liu, Yao, and Wei (see [16]) also introduced a modified nonmonotone BFGS algorithm on the basis of a modified secant condition. Unfortunately, the global convergence result of this method depended on the convex assumption of the objective functions. We note that all these studies concentrate on the case where objective functions are convex (or even uniformly convex). So far, little is known regarding the global convergence of the nonmonotone BFGS method for nonconvex minimization.

In this paper, we investigate the nonmonotone BFGS method in nonconvex minimization. We will propose a modified nonmonotone BFGS method that is globally convergent without a convex assumption on the objective functions. We will show that the global convergence is achieved even if the Armijo line search is used, while in [12,15,16], the Wolfe conditions were used; in [26], a modified Armijo line search was used. The numerical experiments on a set of problems from CUTEr collection indicate that the proposed method is practically effective.

We organize the paper as follows. In the next section, we give the motivation of our study and describe the modified nonmonotone BFGS algorithm. In Section 3, we analyze the convergence of the proposed algorithm. In Section 4, we do some numerical experiments to test the proposed method and compare its performance with the monotone BFGS method. Finally, in Section 5, we draw some conclusions. Throughout the paper, $\|\cdot\|$ denotes the Eucilidean norm of vectors.

2. Algorithm

In this section, we present a modified nonmonotone BFGS method after describing our motivation. The method is based on a modified BFGS method developed in [13]. Li and Fukushima (see [13]) gave a new secant equation

$$B_{k+1}s_k = y_k^* = y_k + t_k s_k, \tag{2.1}$$

with $t_k > 0$ is determined by

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

with the constants $\bar{C} > 0$ and $\mu > 0$. Li and Fukushima replaced all the y_k with y_k^* in (1.4), and obtained the following modified BFGS (MBFGS) update formula

$$B_{k+1} = B_k - \frac{B_k s_k s_k^T B_k}{s_k^T B_k s_k} + \frac{y_k^*(y_k^*)^T}{s_k^T y_k^*}. \tag{2.2}$$

The MBFGS update formula has a nice property that for each k, it always holds that

$$s_k^T y_k^* \geq \bar{C}\|g_k\|^{\mu}\|s_k\|^2 \geq 0,$$

which ensures that B_{k+1} inherits the positive definiteness of B_k. This property is independent of the convexity of f as well as the line search used.

In this paper, we further investigate the global convergence property of the MBFGS. Unlike the traditional line search method [13] or the fixed step length strategy [25], we pay particular attention to the nonmonotone line search strategy.

The traditional line search requires the function value to decrease monotonically at each iteration, namely, $f_{k+1} < f_k$. As a result, it may cause the sequence of iterations following the bottom of a curved narrow valley, which commonly occurred in difficult nonlinear problems (see [5]). To overcome this difficultly, a credible alternative is to allow an occasional increase in the objective function at each iteration.

For easy comprehension of the proposed algorithm, we first briefly recall the earliest nonmonotone line search technique by Grippo, Lampariello, and Lucidi (see [9]). Let $\delta \in (0, 1)$, $\rho \in (0, 1)$, and M_0 be a positive integer. The nonmonotone line search in [9] is to choose the smallest nonnegative integer j_k such as the step length

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

satisfies

$$f(x_k + \alpha_k d_k) \leq \max_{0\leq j\leq m(k)} f(x_{k-j}) + \delta\alpha_k g_k^T d_k, \tag{2.3}$$

where

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

If $m(k) = 0$, the above nonmonotone line search reduces to the Armijo line search.

Based on the above analysis, we are now ready to formally state the steps of the modified BFGS (MBFGS) algorithm with nonmonotone line search, as follows.

Algorithm 2.1. **Step 0.** Choose an initial point $x_0 \in \mathbf{R}^n$ and a symmetric positive definite matrix B_0, a nonnegative integer M_0. Set $k = 0$.
Step 1. If $\|g_k\| = 0$ then stop.
Step 2. For given x_k and B_k, solve $B_k d + g_k = 0$ to obtain a search direction d_k.
Step 3. Determine α_k satisfying (2.3).
Step 4. Set $x_{k+1} = x_k + \alpha_k d_k$. Update B_k to get B_{k+1} according to (2.2).
Step 5. Set $k := k + 1$. Go to Step 1.

3. Convergence analysis

The aim of this section is to study the global convergence behavior of Algorithm 2.1. We first make the following assumptions:

Assumption 3.1. *The level set $\mathcal{L}_0 = \{x | f(x) \le f(x_0)\}$ is bounded.*

Assumption 3.2. *Function f is continuously differentiable on $\mathcal{L}_0$, and there exists a constant $L > 0$ such that*

$$\|g(x) - g(y)\| \le L\|x - y\|, \quad \forall x,\ y \in \mathcal{L}_0. \tag{3.1}$$

Assumption 3.1 and the inequality (2.3) indicate that there exists a subsequence such that $\{f_k\}_K$ is nonincreasing, which ensures that $\{x_k\} \subset \mathcal{L}_0$ and there exists a local minimizer x^* such that

$$\lim_{k\to\infty} f(x_k) = f(x^*).$$

Lemma 3.1. *Let the sequence $\{x_k\}$ be generated by Algorithm 2.1. If $\|g_k\| \ge \epsilon$ holds for all k with some constant $\epsilon > 0$, then there exists a positive constant M, such that*

$$\frac{\|y_k^*\|^2}{s_k^T y_k^*} \le M. \tag{3.2}$$

Proof. It is easy to see that $\{x_k\} \subset \mathcal{L}_0$ and $\|y_k^*\| \le C\|s_k\|$ for some positive constant C. By the definition of t_k, we have

$$s_k^T y_k^* \ge \bar{C}\|g_k\|^{\mu}\|s_k\|^2 \ge \bar{C}\epsilon^{\mu}\|s_k\|^2. \tag{3.3}$$

Hence

$$\frac{\|y_k^*\|^2}{s_k^T y_k^*} \le \frac{C^2\|s_k\|^2}{\bar{C}\epsilon^{\mu}\|s_k\|^2} = M,$$

where $M = \frac{C^2}{\bar{C}\epsilon^{\mu}}$. The proof is complete. □

The following lemma comes from Theorem 2.1 of [2].

Lemma 3.2. *Let the sequence $\{x_k\}$ be generated by Algorithm 2.1. If $\|g_k\| \ge \epsilon$ holds for all k with some constant $\epsilon > 0$, then there are positive constants β_j, $j = 1, 2, 3$, such that, for any k, the inequalities*

$$\|B_i s_i\| \le \beta_1\|s_i\| \quad \text{and} \quad \beta_2\|s_i\|^2 \le s_i^T B_i s_i \le \beta_3\|s_i\|^2 \tag{3.4}$$

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

Lemma 3.3. *Denote*

$$f(x_{h(k)}) = \max_{0\le j\le m(k)} f(x_{k-j}) \quad \text{and} \quad k - m(k) \le h(k) \le k. \tag{3.5}$$

If $f(x_{k+1}) \le f(x_{h(k)})$, $k = 0, 1, \ldots$, then the sequence $\{f(x_{h(k)})\}$ is monotonically nonincreasing. Moreover, we have $x_k \in \mathcal{L}_0$ for all $k \ge 0$.

Proof. By $f(x_{k+1}) \le f(x_{h(k)})$, we have

$$\begin{aligned} f(x_{h(k+1)}) &= \max_{0\le j\le m(k+1)} f(x_{k+1-j}) \\ &\le \max\{\max_{0\le j\le m(k)} f(x_{k-j}), f(x_{k+1})\} \\ &= \max\{f(x_{h(k)}), f(x_{k+1})\} \\ &= f(x_{h(k)}), \end{aligned}$$

which shows that the sequence $\{f(x_{h(k)})\}$ is monotonically nonincreasing. Since $f(x_{h(0)}) = f(x_0)$, we deduce $f(x_k) \le f(x_{h(k-1)}) \le \cdots \le f(x_{h(0)}) = f(x_0)$, $x_k \in \mathcal{L}_0$. $\square$

Lemma 3.4. *If $\|g_k\| \ge \epsilon$ holds for all k with some constant $\epsilon > 0$, then there exists a positive constant $\bar{\alpha}$ such that $\alpha_i \ge \bar{\alpha}$ for all $i \in J = \{i|(3.4)$ holds$\}$.*

Proof. It suffices to consider the case $\alpha_i \ne 1$. Taking into account that $m(k+1) \le m(k)+1$, by the line search rule, we see that the scalar $\widehat{\alpha_i} \equiv \frac{\alpha_i}{\rho}$ does not satisfy (2.3), i.e.,

$$\begin{aligned} f(x_{h(k)} + \hat{\alpha}_{h(k)} d_{h(k)}) &> \max_{0\le j\le m(h(k))} f(x_{h(k)-j}) + \delta\hat{\alpha}_{h(k)} g_{h(k)}^T d_{h(k)} \\ &= f(x_{h(h(k))}) + \delta\hat{\alpha}_{h(k)} g_{h(k)}^T d_{h(k)} \\ &\ge f(x_{h(k)}) + \delta\hat{\alpha}_{h(k)} g_{h(k)}^T d_{h(k)}. \end{aligned} \tag{3.6}$$

By the mean-value theorem and (3.1), we have

$$\begin{aligned} f(x_{h(k)} + \hat{\alpha}_{h(k)} d_{h(k)}) &= f(x_{h(k)}) + \hat{\alpha}_{h(k)} g(x_{h(k)})^T d_{h(k)} + \hat{\alpha}_{h(k)}[g(x_{h(k)} + \theta\hat{\alpha}_{h(k)} d_{h(k)}) - g(x_{h(k)})^T] d_{h(k)} \\ &\le f(x_{h(k)}) + \hat{\alpha}_{h(k)} g(x_{h(k)})^T d_{h(k)} + (\hat{\alpha}_{h(k)})^2 L\|d_{h(k)}\|^2, \end{aligned}$$

where $\theta \in (0, 1)$. Substituting this into (3.6), we get

$$\begin{aligned} \hat{\alpha}_{h(k)} L\|d_{h(k)}\|^2 &> -(1-\delta) g_{h(k)}^T d_{h(k)} \\ &= (1-\delta) d_{h(k)}^T B_{h(k)} d_{h(k)} \\ &\ge (1-\delta)\beta_2\|d_{h(k)}\|^2, \end{aligned}$$

where the last inequality follows from (3.4). The last inequality yields $\hat{\alpha}_{h(k)} \ge ((1-\delta)/L)\beta_2$ and hence

$$\alpha_{h(k)} = \widehat{\alpha}_{h(k)}\rho \ge \frac{(1-\delta)\beta_2\rho}{L} = \bar{\alpha},$$

which shows the desired conclusion. $\square$

Lemma 3.5. *Let $\{x_k\}$ be generated by Algorithm 2.1. If $\|g_k\| \ge \epsilon$ holds for all k with some constant $\epsilon > 0$, then there exists a positive constant M_1, such that*

$$tr(B_{k+1}) < M_1(k+1) \quad \text{and} \quad \sum_{i=0}^{k} \frac{\|B_i s_i\|^2}{s_i^T B_i s_i} < M_1(k+1), \tag{3.7}$$

where $tr(B)$ denotes the trace of matrix B.

Proof. Omitted. $\square$

Based on the above two lemmas, in a way similar to the proof of Lemma 3.3 of [13], it is not difficult to verify the following lemma.

Lemma 3.6. *If $\delta \in [0, 1]$, there is a constant $c > 0$ such that for all $k \ge 1$, we have*

$$\prod_{j=1}^{k} a_j \ge c^k. \tag{3.8}$$

The proof of the following lemma is similar to the one given in [13].

Lemma 3.7. *We have*

$$\sum_{k=0}^{\infty} -\min_{0\le j\le m(k)} g_{k+m(k)-j}^T s_{k+m(k)-j} < \infty. \tag{3.9}$$

Proof. Let $\rho_k = -\delta\alpha_k g_k^T d_k$, by Lemma 3.3 we have

$$\begin{aligned}
&f(x_{k+1}) \le f(x_{h(k)}) - \rho_k,\\
&f(x_{k+2}) \le f(x_{h(k+1)}) - \rho_{k+1} \le f(x_{h(k)}) - \rho_{k+1},\\
&\dots \qquad \dots\\
&f(x_{k+m(k)+1}) \le f(x_{h(k+m(k))}) - \rho_{k+m(k)} \le f(x_{h(k)}) - \rho_{k+m(k)},
\end{aligned}$$

which shows

$$\begin{aligned}
f(x_{h(k+m(k)+1)}) &= \max_{0\le j\le m(k)} f(x_{k+m(k)+1-j})\\
&\le f(x_{h(k)}) - \min_{0\le j\le m(k)} \rho_{k+m(k)-j}, \quad k = 0, 1, \dots.
\end{aligned}$$

Selecting $k = 0, m(k) + 1, \dots, (n - 1)(m(k) + 1)$ in the last inequality respectively, we get

$$\begin{aligned}
&f(x_{h(m(k)+1)}) - f(x_{h(0)}) \le -\min_{0\le j\le m(k)} \rho_{m(k)-j},\\
&f(x_{h(2m(k)+2)}) - f(x_{h(m(k)+1)}) \le -\min_{0\le j\le m(k)} \rho_{2m(k)-j},\\
&\dots \qquad \dots\\
&f(x_{h(nm(k)+n)} - f(x_{h((n-1)(m(k)+1))}) \le -\min_{0\le j\le m(k)} \rho_{nm(k)+n-1-j}.
\end{aligned}$$

Summing up the above n inequalities, we get

$$f(x_{h(nm(k)+n)}) - f(x_{h(0)}) \le -\sum_{k=1}^{n-1} \min_{0\le j\le m(k)} \rho_{k+m(k)-j}. \tag{3.10}$$

From Lemma 3.3, we know that $x_{h(nm(k)+n)} \in \mathcal{L}_0$ and that the sequence $\{f(x_{h(k)})\}$ is monotonically nonincreasing. Therefore, we obtain

$$\begin{aligned}
&\lim_{k\to\infty} f(x_{h(nm(k)+n)}) > -\infty,\\
&\sum_{k=0}^{\infty} \min_{0\le j\le m(k)} \rho_{k+m(k)-j} < +\infty,
\end{aligned}$$

which shows (3.9). □

Now we establish the global convergence theorem of Algorithm 2.1.

Theorem 3.1. *Let $\{x_k\}$ be generated by Algorithm 2.1. Then we have*

$$\liminf_{k\to\infty} \|g_k\| = 0. \tag{3.11}$$

Proof. Lemma 3.7 shows that there is an infinite index set $\mathcal{J}$, such that for all $r \in \mathcal{J}$

$$\sum_r (-g_r^T s_r) < \infty.$$

For the sake of contradiction, we suppose on the contrary that $\|g_r\| \ge \epsilon$ hold for all r with some $\epsilon > 0$. Then, we have

$$\begin{aligned}
\infty &> \sum_{r\in\mathcal{J}} (-g_r^T s_r)\\
&= \sum_{r\in\mathcal{J}} \|g_r\|^2 \alpha_r \frac{s_r^T B_r s_r}{\|B_r s_r\|^2}\\
&\ge \epsilon^2 \sum_{r\in\mathcal{J}} \alpha_r \frac{s_r^T B_r s_r}{\|B_r s_r\|^2}\\
&\ge \epsilon^2 \bar{\alpha} \sum_{r\in\mathcal{J}} \frac{s_r^T B_r s_r}{\|B_r s_r\|^2}.
\end{aligned}$$

Therefore, for any $\zeta > 0$, there exists an integer $r_0 > 0$, such that for any positive integer q,

$$q\left(\prod_{r=r_0+1, r\in\mathcal{J}}^{r_0+q} \alpha_r \frac{s_r^T B_r s_r}{\|B_r s_r\|^2}\right)^{1/q} \le \sum_{r=r_0+1, r\in\mathcal{J}}^{r_0+q} \alpha_r \frac{s_r^T B_r s_r}{\|B_r s_r\|^2} \le \zeta,$$

where the first inequality follows from the geometric inequality. Thus

$$\begin{aligned}\left(\prod_{r=r_0+1, r\in\mathcal{J}}^{r_0+q}\alpha_r\right)^{1/q} &\le \frac{\zeta}{q}\left(\prod_{r=r_0+1, r\in\mathcal{J}}^{r_0+q}\frac{\|B_r s_r\|^2}{s_r^T B_r s_r}\right)^{1/q}\\ &\le \frac{\zeta}{q^2}\sum_{r=r_0+1, r\in\mathcal{J}}^{r_0+q}\frac{\|B_r s_r\|^2}{s_r^T B_r s_r}\\ &\le \frac{\zeta}{q^2}\sum_{r=0, r\in\mathcal{J}}^{r_0+q}\frac{\|B_r s_r\|^2}{s_r^T B_r s_r}\\ &\le \frac{\zeta(r_0+q+1)}{q^2}M_1.\end{aligned}$$

Letting $q \to \infty$, the last inequality yields a contradiction, because Lemma 3.6 ensures that the left-hand side of the above inequality is greater than a positive constant. Thus we get (3.11). □

4. Numerical experiments

In this section, we examine the numerical behavior of Algorithm 2.1, which we call MBFGS, on a set of test problems. The algorithm was coded in Fortran77 and in double precision arithmetic. All the experiments were run on a PC with CPU Intel Pentium Dual E2140 1.6GHz, 512M bytes of SDRAM memory, and Red Hat Linux 9.03 operating system. For each test problem, the termination condition is

$$\|g(x_k)\| \le 10^{-5}. \tag{4.1}$$

The program was also terminated if the number of iterations exceed 10000 or the number of function evaluations reaches 20 000. Our experiments were performed on the 30 typical nonlinear unconstrained problems from CUTEr (see [4]) collection. Each selected problem is regular, that is, its first and second derivatives exist and are continuous everywhere. We tested the algorithm on each problem with different dimensions, i.e., 100, 500 and 1000 dimensions. That is to say, there are 90 test problems in total.

For MBFGS method, we set the initial matrix $B_0 = I$, i.e., the identify matrix. We chose initial parameters $\rho = 0.29$, $\delta = 10^{-1}$ in the line search. The parameters of the MBFGS update are specified as follows: we set $\mu = 4$, the chosen $\bar{C} = 10^{-2}$ if $\|g_k\| \le 10^{-2}$, and $\bar{C} = 0$, otherwise. After running a few problems with different values of $M_0 \in \{3, 20\}$, we found $M_0 = 5$ always leads to better numerical behavior. To determine the search direction d_k, we used the linear conjugate gradient method to solve the linear system (1.3). The linear conjugate gradient method can be seen in [18] for more detail. In order to assess the reliability of MBFGS method, we also tested this method with parameter $M_0 = 1$. Tables A.1 and A.2 in the Appendices report the performance of MBFGS with different M_0. The columns have the following meaning:

Problem:	name of the problem;
Dim:	dimension of the problem;
Iter:	number of iterations;
Fcnt:	number of function evaluations;
Time:	CPU time in seconds;
Fv:	final function value;
Norm1:	maxim-norm of the final gradient;
Norm2:	2-norm of the final gradient.

The performance of both algorithms is evaluated by the use of the profiles of Dolan and Moré (see [8]). In Figs. 1–3, we display the performance profiles of Dolan and Moré for $M_0 = 1$ and $M_0 = 5$ referring to number of iterations, number of function and gradient evaluations, and CPU time, respectively.

In this series of experiments, MBFGS method with $M_0 = 5$ can solve over 95% of the test problems, while with $M_0 = 1$ solve nearly 75%. This can be observed at the right of the three figures where $\tau = 5$. Observing all these figures, we conclude that MBFGS method with $M_0 = 5$ is always the top performer for all values of τ, which shows that MBFGS method with $M_0 = 5$ performs better than with $M_0 = 1$ does, and requires less iterations, less function, gradient evaluations, and little time consumption. Moreover, preliminary experimental comparisons also indicate that the nonmonotone line search strategy is very beneficial to the performance.

5. Conclusions

In this paper, we developed a nonmonotone BFGS method for solving unconstrained optimization problems. An important feature of the proposed method is that the function value at each iteration allows for an occasional increase. Compared with some extant methods in this literature, our method can converge to a local optimal point without a convex

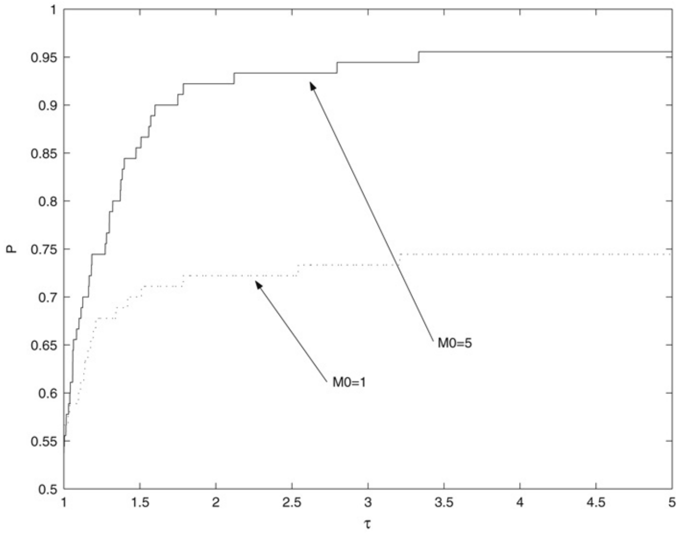

Fig. 1. Performance profiles based on iterations.

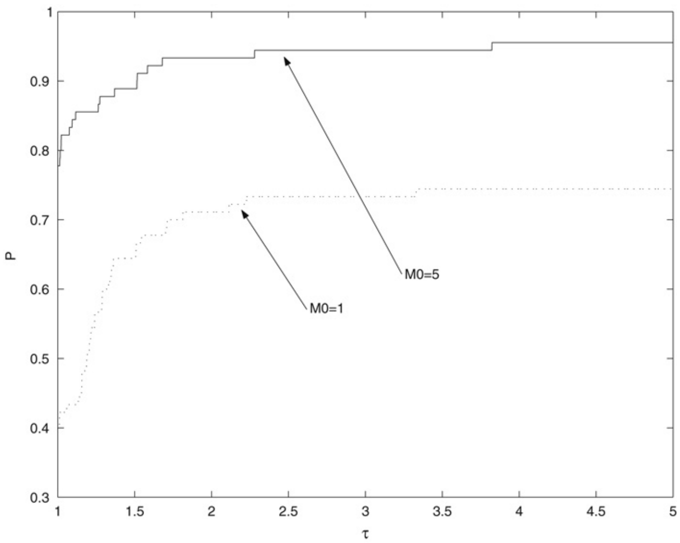

Fig. 2. Performance profiles based on function evaluations.

assumption on the objective functions. Additionally, the method can be considered as an extension of the method in [13] to the non monotone scheme. We also tested our method on a large number of problems from CUTEr library, which indicates our method is promising. Lately, [16] has already proven the superlinear convergence when using Wolfe condition. Although we all suppose that superlinear convergence holds as for the Armijo line search condition, at present we cannot give a complete proof of it yet. This may be a part of our further research.

Acknowledgments

We are very thankful Professor Dong-Hui Li for careful reading and his constructive criticisms on this paper. We are also very grateful to two anonymous referees for their useful suggestions and comments on the previous version of this paper.

Appendix

See Tables A.1 and A.2.

Table A.1
Test results for MBFGS method with $M_0 = 5$.

Problem	Dim	Iter	Fcnt	Time	Fv	Norm1	Norm2
BDQRTIC	100	443	1915	1.656	0.378769E+03	0.250392E−05	0.915550E−05
	500	901	2378	76.969	0.198103E+04	0.242793E−05	0.953485E−05
	1000	1256	2930	352.781	0.398381E+04	0.295526E−05	0.412854E−05
TRIDIA	100	1168	2261	2.203	0.620690E−12	0.242948E−05	0.800859E−05
	500	8172	11881	452.312	0.104949E−10	0.453505E−05	0.936935E−05
	1000	10001	12115	>999	0.160228E−04	0.562001E−02	0.985507E−02
ARWHEAD	100	18	52	0.016	0.000000E+00	0.718305E−08	0.571529E−07
	500	8	42	0.312	0.000000E+00	0.174428E−05	0.174679E−05
	1000	2498	20000	373.078	0.210731E−11	0.724792E−05	0.101897E−04
NONDIA	100	16	47	0.047	0.278850E−14	0.797379E−05	0.810409E−05
	500	18	48	0.672	0.145978E−17	0.126992E−06	0.127327E−06
	1000	21	48	3.109	0.459945E−10	0.537381E−06	0.860412E−05
DQDRTIC	100	360	464	0.469	0.116226E−13	0.221609E−05	0.228409E−05
	500	22	145	2.656	0.141824E−13	0.168659E−05	0.229006E−05
	1000	18	141	7.109	0.307336E−13	0.205146E−05	0.291868E−05
EG2	100	1060	1151	1.641	−0.994474E+02	0.100968E−05	0.999565E−05
	500	10001	10095	162.406	−0.499500E+03	0.542308E−04	0.542308E−04
	1000	8739	8771	>999	−0.999000E+03	0.631879E−05	0.658330E−05
DIXMAANA	100	21	75	0.031	0.100000E+01	0.122078E−06	0.994257E−06
	500	22	38	0.797	0.100000E+01	0.200504E−06	0.286111E−05
	1000	25	57	3.781	0.100000E+01	0.272668E−06	0.497712E−05
DIXMAANB	100	38	190	0.078	0.100000E+01	0.139711E−05	0.232233E−05
	500	53	302	2.969	0.100000E+01	0.178186E−05	0.246544E−05
	1000	18	22	2.703	0.100000E+01	0.189099E−05	0.981745E−05
DIXMAANC	100	26	268	0.078	0.100000E+01	0.469154E−06	0.828588E−06
	500	100	466	5.984	0.100000E+01	0.260906E−05	0.564495E−05
	1000	53	241	13.953	0.100000E+01	0.508706E−06	0.131216E−05
DIXMAANE	100	149	865	0.484	0.100000E+01	0.448115E−05	0.882760E−05
	500	572	1997	49.438	0.100000E+01	0.710376E−05	0.982050E−05
	1000	1639	3258	382.516	0.100000E+01	0.917024E−05	0.960350E−05
EDENSCH	100	95	542	0.281	0.603284E+03	0.664234E−05	0.793662E−05
	500	107	813	13.438	0.300324E+04	0.144101E−05	0.438526E−05
	1000	73	421	25.469	0.600328E+04	0.602919E−05	0.777491E−05
VARDIM	100	33	59	0.031	0.872453E−18	0.186810E−06	0.108666E−05
	500	46	82	1.562	0.486382E−28	0.177635E−14	0.139482E−13
	1000	51	91	6.281	0.586471E−25	0.270894E−13	0.484343E−12
LIARWHD	100	22	58	0.031	0.498248E−14	0.105759E−05	0.105969E−05
	500	6040	6055	232.969	0.211344E−15	0.243754E−06	0.243865E−06
	1000	23	58	3.484	0.613830E−13	0.393789E−05	0.400780E−05
DIXMAANF	100	282	711	0.500	0.100000E+01	0.827457E−05	0.828989E−05
	500	378	20000	124.969	0.569488E+02	0.466753E+03	0.585048E+03
	1000	1007	3057	320.203	0.100000E+01	0.801350E−05	0.989246E−05
DIXMAANG	100	187	916	0.625	0.100000E+01	0.424353E−05	0.756939E−05
	500	493	1649	41.859	0.100000E+01	0.474776E−05	0.663421E−05
	1000	759	2680	279.250	0.100000E+01	0.822893E−05	0.928678E−05
DIXMAANI	100	221	1026	0.688	0.100000E+01	0.934968E−05	0.961035E−05
	500	606	2282	57.500	0.100000E+01	0.729596E−05	0.999226E−05
	1000	727	2644	245.734	0.100000E+01	0.889362E−05	0.960496E−05
DIXMAANJ	100	256	804	0.578	0.100000E+01	0.331778E−05	0.636061E−05
	500	826	20000	44.703	0.100017E+01	0.318389E−02	0.388541E−02
	1000	819	20000	148.484	0.998927E+00	0.192729E−02	0.214863E−02
DIXMAANK	100	341	1383	1.031	0.100000E+01	0.506463E−05	0.851928E−05
	500	650	1547	43.438	0.100000E+01	0.620581E−05	0.929595E−05
	1000	1079	2977	341.703	0.100000E+01	0.638831E−05	0.803693E−05
ENGVAL1	100	74	259	0.125	0.109088E+03	0.598266E−05	0.629907E−05
	500	52	57	2.078	0.553135E+03	0.501439E−05	0.856073E−05
	1000	73	587	37.844	0.110819E+04	0.162387E−05	0.389614E−05

Table A.1 (*continued*)

Problem	Dim	Iter	Fcnt	Time	Fv	Norm1	Norm2
COSINE	100	356	1479	1.375	−0.980481E+02	0.333354E−05	0.559943E−05
	500	689	20000	14.625	−0.416107E+03	0.761722E+02	0.945065E+02
	1000	668	20000	49.375	−0.991941E+03	0.735868E+02	0.125265E+03
DENSCHNB	100	11	15	0.016	0.173967E−10	0.117968E−05	0.834217E−05
	500	13	17	0.453	0.828555E−11	0.364100E−06	0.575692E−05
	1000	13	17	1.750	0.165711E−10	0.364100E−06	0.814152E−05
DENSCHNF	100	17	50	0.016	0.146395E−16	0.964512E−08	0.735622E−07
	500	39	54	1.391	0.365620E−14	0.512373E−07	0.974303E−06
	1000	18	72	2.688	0.409615E−12	0.285547E−06	0.882188E−05
SINQUAD	100	7945	8022	12.844	0.731023E−06	0.101213E−05	0.999933E−05
	500	10001	10053	529.250	0.918776E−04	0.414615E−05	0.925876E−04
	1000	10001	10051	>999	0.813984E−04	0.188281E−05	0.594997E−04
PENALTY1	100	11	17	0.016	0.122130E−12	0.698943E−07	0.698943E−06
	500	19	26	0.578	0.687326E−14	0.741526E−08	0.165810E−06
	1000	16	24	2.031	0.374276E−13	0.122356E−07	0.386924E−06
SROSENBR	100	55	90	0.078	0.185677E−12	0.110436E−05	0.856202E−05
	500	64	98	2.500	0.112077E−14	0.331479E−07	0.572612E−06
	1000	82	117	12.578	0.126350E−14	0.229630E−07	0.589323E−06
WOODS	100	70	165	0.125	0.135795E−13	0.854810E−06	0.494393E−05
	500	105	196	5.391	0.283645E−13	0.329604E−06	0.442147E−05
	1000	90	185	19.281	0.329270E−13	0.307055E−06	0.717061E−05
DQRTIC	100	447	520	0.578	0.156084E−06	0.101146E−05	0.998635E−05
	500	1024	1077	39.125	0.263325E−06	0.655215E−06	0.999791E−05
	1000	1818	2050	283.766	0.338931E−06	0.316444E−06	0.999681E−05
LIARUHD	100	22	58	0.016	0.498248E−14	0.105759E−05	0.105969E−05
	500	6040	6055	230.781	0.211354E−15	0.243761E−06	0.243872E−06
	1000	23	58	3.531	0.613632E−13	0.393802E−05	0.400792E−05
BROWNAL	100	6	12	0.000	0.413742E−12	0.273347E−06	0.273347E−05
	500	7	15	0.234	0.110736E−18	0.547393E−10	0.122400E−08
	1000	6	14	0.766	0.684817E−16	0.113504E−08	0.358934E−07
GENHVMPS	100	283	1169	0.891	0.328789E−10	0.135312E−05	0.333979E−05
	500	753	20000	31.875	0.278751E+03	0.469849E+01	0.239082E+02
	1000	781	3247	338.953	0.201980E−09	0.293563E−05	0.893456E−05

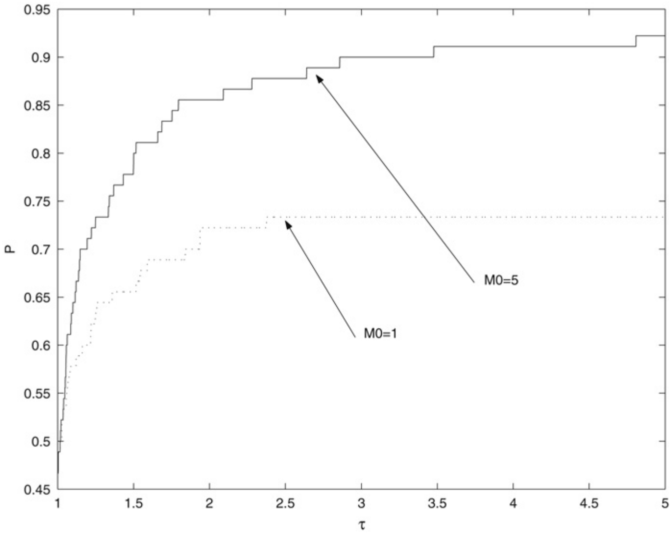

Fig. 3. Performance profiles based on CPU time.

Table A.2
Test results for MBFGS method with $M_0 = 1$.

Problem	Dim	Iter	Fcnt	Time	Fv	Norm1	Norm2
BDQRTIC	100	1126	2473	2.016	0.378769E+03	0.973968E−05	0.974081E−05
	500	2343	20000	109.484	0.198101E+04	0.121005E−03	0.134010E−03
	1000	1709	20000	331.562	0.398381E+04	0.146151E−03	0.191182E−03
TRIDIA	100	2086	7527	5.234	0.128268E−12	0.338185E−05	0.991499E−05
	500	10001	12197	414.875	0.458128E−06	0.950204E−03	0.155850E−02
	1000	10001	12521	>999	0.513846E−05	0.317798E−02	0.688690E−02
ARWHEAD	100	17	70	0.016	0.000000E+00	0.244400E−06	0.260614E−06
	500	8	42	0.297	0.000000E+00	0.174428E−05	0.174679E−05
	1000	2348	20000	353.797	0.118675E−10	0.212200E−04	0.271238E−04
NONDIA	100	10	43	0.031	0.401461E−14	0.329193E−05	0.331725E−05
	500	13	47	0.469	0.208683E−11	0.230949E−06	0.258671E−05
	1000	12	43	1.844	0.660214E−12	0.326649E−07	0.103246E−05
DQDRTIC	100	941	20000	0.812	0.944345E−02	0.874649E+00	0.125436E+01
	500	16	143	2.125	0.127235E−13	0.171476E−05	0.222458E−05
	1000	16	163	5.328	0.939376E−13	0.346238E−05	0.677983E−05
EG2	100	719	20000	1.094	−0.978252E+02	0.587348E+00	0.698602E+00
	500	125	240	5.641	−0.499500E+03	0.622648E−07	0.623022E−07
	1000	8735	8773	>999	−0.999000E+03	0.578825E−06	0.610812E−06
DIXMAANA	100	18	91	0.031	0.100000E+01	0.118197E−05	0.992872E−05
	500	14	24	0.531	0.100000E+01	0.817323E−08	0.840017E−07
	1000	14	25	2.156	0.100000E+01	0.498075E−06	0.909235E−05
DIXMAANB	100	627	20000	0.844	0.977192E+02	0.460268E+01	0.306453E+02
	500	25	79	1.125	0.100000E+01	0.802093E−06	0.307210E−05
	1000	17	22	2.609	0.100000E+01	0.627185E−07	0.410201E−06
DIXMAANC	100	27	177	0.047	0.100000E+01	0.148033E−05	0.584972E−05
	500	30	366	2.625	0.100000E+01	0.308459E−07	0.708127E−07
	1000	34	176	10.406	0.100000E+01	0.383686E−06	0.130185E−05
DIXMAANE	100	225	1472	0.891	0.100000E+01	0.649432E−05	0.734608E−05
	500	440	1580	27.531	0.100000E+01	0.659473E−05	0.874774E−05
	1000	1577	3186	335.953	0.100000E+01	0.698485E−05	0.905067E−05
EDENSCH	100	68	721	0.297	0.603285E+03	0.285512E−05	0.756254E−05
	500	71	484	4.703	0.300328E+04	0.598739E−05	0.811378E−05
	1000	70	486	28.562	0.600328E+04	0.484030E−05	0.628654E−05
VARDIM	100	33	59	0.047	0.872453E−18	0.186810E−06	0.108663E−05
	500	46	82	1.438	0.486382E−28	0.177635E−14	0.139486E−13
	1000	51	91	6.250	0.586471E−25	0.270894E−13	0.484343E−12
LIARWHD	100	22	58	0.031	0.498248E−14	0.105759E−05	0.105969E−05
	500	28	63	1.078	0.584954E−15	0.454329E−08	0.491337E−07
	1000	23	58	3.469	0.613830E−13	0.393789E−05	0.400780E−05
DIXMAANF	100	634	20000	0.906	0.211244E+01	0.123092E+01	0.177252E+01
	500	635	20000	18.594	0.101670E+01	0.217442E−01	0.643013E−01
	1000	904	3538	291.000	0.100000E+01	0.781562E−05	0.999061E−05
DIXMAANG	100	589	20000	0.875	0.237062E+02	0.131999E+01	0.952375E+01
	500	547	2226	37.312	0.100000E+01	0.880322E−05	0.971855E−05
	1000	1019	4049	340.859	0.100000E+01	0.808716E−05	0.998680E−05
DIXMAANI	100	256	1861	1.094	0.100000E+01	0.415072E−05	0.677571E−05
	500	606	2776	50.219	0.100000E+01	0.709648E−05	0.980131E−05
	1000	879	3407	265.953	0.100000E+01	0.753716E−05	0.998771E−05
DIXMAANJ	100	255	1212	0.672	0.100000E+01	0.166855E−05	0.532523E−05
	500	625	20000	12.859	0.180354E+03	0.147128E+02	0.621093E+02
	1000	597	20000	140.391	0.930437E+00	0.119547E+00	0.224119E+00
DIXMAANK	100	268	1778	0.984	0.100000E+01	0.717222E−05	0.825474E−05
	500	776	3269	67.047	0.100000E+01	0.878809E−05	0.978940E−05
	1000	1228	5094	429.906	0.100000E+01	0.891589E−05	0.988304E−05
ENGVAL1	100	76	577	0.156	0.109088E+03	0.244684E−05	0.836689E−05
	500	57	68	2.156	0.553135E+03	0.821676E−05	0.990785E−05
	1000	57	387	18.094	0.110819E+04	0.650798E−05	0.826637E−05

Table A.2 (*continued*)

Problem	Dim	Iter	Fcnt	Time	Fv	Norm1	Norm2
COSINE	100	665	20000	1.125	−0.943091E+02	0.553901E+01	0.107270E+02
	500	668	20000	14.344	−0.367893E+03	0.178876E+03	0.215889E+03
	1000	2144	3078	424.516	−0.999000E+03	0.399999E−05	0.733773E−05
DENSCHNB	100	10	17	0.016	0.713042E−11	0.676780E−06	0.584310E−05
	500	11	18	0.406	0.617995E−11	0.284354E−06	0.540604E−05
	1000	12	20	1.781	0.105443E−12	0.302717E−07	0.806764E−06
DENSCHNF	100	20	62	0.031	0.464442E−13	0.718744E−06	0.512993E−05
	500	30	67	1.016	0.952217E−13	0.257523E−06	0.440588E−05
	1000	17	98	2.547	0.150612E−18	0.441879E−09	0.992700E−08
SINQUAD	100	10001	10079	16.531	0.650696E−05	0.269102E−05	0.267067E−04
	500	10001	10074	520.734	0.751477E−04	0.354325E−05	0.791072E−04
	1000	10001	10076	>999	0.881418E−04	0.200399E−05	0.633293E−04
PENALTY1	100	11	17	0.000	0.122130E−12	0.698943E−07	0.698943E−06
	500	19	26	0.594	0.687326E−14	0.741526E−08	0.165810E−06
	1000	16	24	1.953	0.374276E−13	0.122356E−07	0.386924E−06
SROSENBR	100	55	91	0.078	0.124336E−12	0.890749E−06	0.690356E−05
	500	63	99	2.500	0.141194E−14	0.369969E−07	0.639010E−06
	1000	82	141	12.297	0.803981E−15	0.195557E−07	0.477558E−06
WOODS	100	648	20000	0.562	0.542244E+03	0.356118E+02	0.195696E+03
	500	149	303	7.328	0.643719E−13	0.176210E−06	0.250369E−05
	1000	76	172	16.781	0.392707E−14	0.689598E−07	0.137117E−05
DQRTIC	100	441	512	0.484	0.156117E−06	0.100862E−05	0.998486E−05
	500	1159	1280	40.234	0.268673E−06	0.464341E−06	0.999296E−05
	1000	650	20000	49.719	0.417003E+11	0.552532E+07	0.771645E+08
LIARUHD	100	22	58	0.031	0.498248E−14	0.105759E−05	0.105969E−05
	500	28	63	1.078	0.585082E−15	0.454244E−08	0.491391E−07
	1000	23	58	3.531	0.613632E−13	0.393802E−05	0.400792E−05
BROWNAL	100	6	12	0.000	0.413742E−12	0.273347E−06	0.273347E−05
	500	7	15	0.250	0.110736E−18	0.547393E−10	0.122400E−08
	1000	6	14	0.812	0.684817E−16	0.113504E−08	0.358934E−07
GENHVMPS	100	727	20000	3.031	0.149497E+02	0.286501E+01	0.459844E+01
	500	1109	20000	64.297	0.155323E−03	0.320679E−02	0.772203E−02
	1000	672	20000	70.484	0.132927E+08	0.761592E+02	0.238501E+04

References

[1] E.G. Birgin, J.M. Martínez, M. Raydan, Nonmonotone spectral projected gradient methods on convex sets, SIAM J. Optim. 10 (2000) 1196–1121.
[2] 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.
[3] 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.
[4] A.R. Conn, N.I.M. Gould, Ph.L. Toint, CUTE: Constrained and unconstrained testing environment, ACM Trans. Math. Software 21 (1995) 123–160.
[5] Y.H. Dai, On the nonmonotone line search, J. Optim. Theory Appl. 112 (2002) 315–330.
[6] Y.H. D ai, Convergence properties of the BFGS algorithm, SIAM J. Optim. 13 (2002) 693–701.
[7] N.Y. Deng, Y. Xiao, F.J. Zhou, Nonmonotonic trust-region algorithm, J. Optim. Theory Appl. 26 (1993) 259–285.
[8] E.D. Dolan, J.J. Moré, Benchmarking optimization software with performance profiles, Math. Program. 91 (2002) 201–213.
[9] L. Grippo, F. Lampariello, S. Lucidi, A nonmonotone line search technique for Newton's method, SIAM J. Numer. Anal. 23 (1986) 707–716.
[10] L. Grippo, F. Lampariello, S. Lucidi, A truncated Newton method with nonmonotone line search for unconstrained optimization, J. Optim. Theory Appl. 60 (1989) 401–419.
[11] L. Grippo, F. Lampariello, S. Lucidi, A class of nonomonotone stabilization methods in unconstrained optimization, Numer. Math. 59 (1991) 779–805.
[12] J. Han, G. Liu, Global convergence analysis of a new nonmotone BFGS algorithm on convex objective function, Comput. Optim. Appl. 7 (1997) 277–289.
[13] D.H.Li.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.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, D. Sun, Global convergence of the BFGS algorithm with nonmonotone linesearch, Optim. 34 (1995) 147–159.
[16] 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. doi:10.1016/j.cam.2007.08.017.
[17] W.F. Mascarenhas, The BFGS method with exact line searches fails for non-convex objective functions, Math. Program. 99 (2004) 49–61.
[18] J. Nocedal, S.J. Wright, Numerical Optimization, Springer, New York, 1999.
[19] E. Panier, A. Tits, Avoiding the maratos effect by means of a nonmonotone linesearch, I: general constrained problems, SIAM J. Numer. Anal. 28 (1991) 1183–1195.
[20] M.J.D. Powell, Some global convergence properties of a variable metric algorithm for minimization without exact line searches, in: R.W. Cottle, C.E. Lemke (Eds.), Non- linear Programming, SIAM-AMS Proceedings, Vol. LX., American Mathematical Society, Providence, 1976, pp. 53–72.
[21] M. Raydan, The Barzilai and Borwein method for the large scale unconstrained optimization, SIAM J. Optim. 7 (1997) 26–33.
[22] Ph.L. Toint, An assessment of non-monotone line search technique for unconstrained optimization, SIAM J. Sci. Comput. 17 (1996) 725–739.
[23] Ph.L. Toint, A nonmonotone trust-region algorithm for nonlinear optimization subject to convex constraints, Math. Program. 77 (1997) 69–94.

[24] Y. Xiao, Q. Hu, Subspace Barzilai-Borwein gradient method for large-scale bound constrained optimization, Appl. Math. Optim. doi:10.2007/s00245-008-9038-9.
[25] L. Zhang, A globally convergent BFGS method for nonconvex minimization without line searches, Optim. Method. Softw. 20 (2005) 737–747.
[26] W. Zhou, L. Zhang, Global convergence of the nonmonotone MBFGS method for nonconvex unconstrained minimization, J. Comput. Appl. Math., doi:10.1016/j.cam.2007.12.011.
[27] H. Zhang, W.W. Hager, A nonmonotone line search technique and its application to unconstrained optimization, SIAM J. Optim. 14 (2004) 1043–1056.