Fundamental Journal of Mathematics and Mathematical Sciences
Vol. 7, Issue 2, 2017, Pages 99-115
This paper is available online at http://www.frdint.com/
Published online July 2, 2017

A MODIFIED NONMONOTONE MEMORY GRADIENT METHOD FOR UNCONSTRAINED OPTIMIZATION

KE SU[a,*], ZIXING RONG[a] and RUI ZHAO[b]

[a]College of Mathematics and Information Science
Hebei University
Baoding 071002
P. R. China

[b]School of Science
Hebei University of Technology
Tianjin 300401
P. R. China

Abstract

We propose a modified nonmonotone line search technique which includes a convex combination of the maximum objective function value of the preceding successful iterates and the current objective function value. Our algorithm is proved to be well-defined and effective through theoretical analysis and numerical experiments. Compared to other nonmonotone methods, our algorithm makes full use of functional

Keywords and phrases: unconstrained optimization, memory gradient methods, global convergence, nonmonotone.

2010 Mathematics Subject Classification: 90C30.

This work was supported by the National Natural Science Foundation of China (No. 11101115), the Natural Science Foundation of Hebei Province (No. A2014201033), and the Key Research Foundation of Education Bureau of Hebei Province (No. ZD2015069).

*Corresponding author

Received April 20, 2017; Revised April 25, 2017

information and is easier to practice. Its numerical results show that this algorithm is efficient.

1. Introduction

We consider the following unconstrained optimization problem

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

where $f : R^n \to R$ is a continuously differentiable function, R^n is a Euclidean space. This problem, arising often in economy, management, control science and many fields in society, is extremely important. There are various methods to attack problem (1). Almost all the methods for problem (1) are the iterative ones and they generate a sequence $\{x_k\}$ converging to the desired solution. The iteration has the form

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

where d_k is the current direction and $\alpha_k > 0$ is a step length to reduce the objective function $f(x)$.

Among all the effective methods, conjugate gradient methods [1] and memory gradient methods [2] are two powerful methods for solving large scale unconstrained optimization problems, because they avoid the computation and storage of some matrices associated with Newton type methods. Many authors presented the different kinds of formulas to calculate the search direction d_k in conjugate gradient methods, such as Fletcher-Reeves (FR), Polark-Ribire-Polyak (PRP), Polark-Ribire-Polyak plus $(\text{PRP}+)$ and Dai-Yuan (DY), etc. Moreover, the global convergence properties of conjugate gradient methods have been studied by many researchers [3-5]. But different from the conjugate gradient appraoches, there are no uniform formula of d_k for memory gradient methods which was proposed by Miele and Cantrell [2] and Cragg and Levy [6]. Moreover, memory gradient methods sufficiently use the previous multi-step iterative information at every iteration and add the freedom of some parameters to guarantee quick convergent and robust. Many authors have studied the memory gradient method [1, 12-14] and their global convergence properties for these methods and yielded substantial results [7-11].

Motivated by the above idea, in this paper, we proposed a modified nonmonotone memory gradient method, different from those above, we replace the Armijo monotone line search by our modified nonmonotone line search technique. The stepsize determined by Armijo monotone line search may considerably slow the rate of convergence in the presence of the narrow curved valley. But nonmonotone line search may solve this drawback in a certain degree. Based on the previous reference [16], we present a modified nonmonotone memory gradient method via combining nonmonotone line search by Yu and Pu [15] with monotone Armijo line search so that the best convergence result can be obtained in case that α_k is chosen by nonmonotone when the iterates were far from the optimum, and α_k is chosen by monotone when the iterates were near an optimum. Our method performs efficiently both in theory and numerical results. The search direction that generated by the new method automatically satisfies sufficient descent condition at every iteration without requiring conditions. Furthermore, we prove that the new method with nonmonotone line search rule is globally convergent under mild conditions. Numerical experiments show that the proposed method is efficient.

This paper is organized as follows. In Section 2, we introduce a modified nonmonotone line search which includes a convex combination of the maximum function value of some preceding successful iterates and the current function value, then the new algorithm follows. In Section 3, we establish the global convergence of the algorithms. Some numerical experiments are given in Section 4.

2. A Modified Nonmonotone Memory Gradient Method

In monotonic method, the step size α_k along the search direction d_k is determined by

$$f(x_k + \alpha_k d_k) = \min_{\alpha > 0} f(x_k + \alpha d_k).$$

In fact, the exact step size is difficult or even impossible to seek in practical computation, and thus many researchers constructed some inexact line search rule, such as Armijo rule, Goldstein rule, Wolfe rule and nonmontone line search [17].

Chamberlain et al. [18] proposed a watchdog technique for constrained optimization in 1982, in which some standard line search conditions were relaxed to overcome the Marotos effect. Motivated by this idea, Grippo et al. [17] presented a nonmonotone Armijo-type line search technique for the Newton method. The traditional line search rules require the function value descent monotonically at each iteration. It may considerably slow the rate of convergence in the intermediate stages of the minimization process, especially in the presence of the narrow curved valley. However, the nonmonotone line search rules are effective or even powerful at some iteration, especially when the iterates are trapped in a narrow curved valley of objective functions.

The earliest nonmonotone line search framework was developed by Grippo et al. [17] for Newton-type methods. Due to its excellent numerical exhibition, many nonmonotone techniques have been developed in recent years, before introducing the new nonmonotone technique, we describe the nonmonotone Armijo rule first. Let α_k be a stepsize with $\alpha_k \geq 0$ and d_k be the search direction satisfied $g_k^T d_k \leq 0,$ given $a > 0, \gamma \in (0, 1), \beta \in (0, 1)$ and M be a nonnegative integer. For each k, let $m(k)$ satisfies

$$m(0) = 0, \quad 0 \leq m(k) \leq \min[m(k-1) + 1, M], \quad \text{for } k \geq 1.$$

Let $\alpha_k = \beta^{p^k} a$ and p^k be the smallest nonnegative integer p such that

$$f(x_k + \alpha_k d_k) \leq \max_{0 \leq j \leq m(k)} [f(x_{k-j})] + \gamma \beta^{p^k} a g_k^T d_k.$$

If the search is the nonmonotone Goldstein line search, then α_k should satisfy the following condition:

$$f(x_k + \alpha_k d_k) \leq \max_{0 \leq j \leq m(k)} [f(x_{k-j})] + \mu_1 \lambda_k g_k^T d_k,$$

$$f(x_k + \alpha_k d_k) \leq \max_{0 \geq j \leq m(k)} [f(x_{k-j})] + \mu_2 \lambda_k g_k^T d_k,$$

where $0 < \mu_1 \leq \mu_2 < 1$.

If the search is the nonmonotone Wolfe line search, then α_k should satisfy the following condition:

$$f(x_k + \alpha_k d_k) \leq \max_{0 \leq j \leq m(k)} [f(x_{k-j})] + \gamma_1 \alpha_k g_k^T d_k,$$

$$g(x_k + \alpha_k d_k)^T d_k \geq \gamma_2 g_k^T d_k,$$

where $0 < \gamma_1 \leq \gamma_2 < 1$.

Since Grippo et al. [17] proposed a nonmonotone line search rule for Newton method, the nonmonotone line search methods have been studied by many authors, e.g., Toint [19], Dai [20], Zhang and Hager [16], Shi and Shen [21], Yu and Pu [15]. Theoretical analysis and numerical results show that the nonmonotone algorithms are very efficient.

Although these nonmonotone techniques work well in many cases, there are some drawbacks. First, a good function value generated in any iteration is essentially discarded due to the maximum. Second, the numerical performance is very much dependent on the choice of M [17, 19, 22] in some cases. In this paper, we adopt the nonmonotone line search proposed by Yu and Pu [15] as follows

$$\lambda_{kr} \geq \lambda, \quad r = 0, 1, 2, \cdots, m(k) - 1, \quad \sum_{r=0}^{m(k)-1} \lambda_{kr} = 1,$$

where $\lambda \in (0, 1]$, $M \geq 1$ is a positive integer, $m(k) = \min[k + 1, M]$.

Let $\alpha_k = \beta^{p_k}$ be bounded above and satisfy:

$$f(x_k + \alpha_k d_k) \leq \max\left[f(x_k), \sum_{r=0}^{m(k)-1} \lambda_{kr} f(x_{k-r})\right] + \gamma \alpha_k \langle d_k, g(x_k) \rangle. \tag{2}$$

As well known, the best convergence results were obtained α_k is chosen by nonmonotone when the iterates were far from the optimum, and α_k is chosen by monotone when the iterates were near an optimum. So we give a modified nonmonotone line search which is a convex combination of monotone line search and nonmonotone line search by Yu and Pu [15], when the value of μ is closer to 1, the modified line search closely approximates the usual monotone line search, and as μ approaches 0, the scheme becomes more nonmonotone. The modified line search is as following:

Let $\lambda \in (0, 1]$, $M \geq 1$ is a positive integer, defined $m(k) = \min[k + 1, M]$

$$\lambda_{kr} \geq \lambda, \quad r = 0, 1, 2, \cdots, m(k) - 1, \quad \sum_{r=0}^{m(k)-1} \lambda_{kr} = 1.$$

Let $\alpha_k = \beta^{p_k}$ be bounded above and satisfy:

$$f(x_k + \alpha_k d_k) \leq \mu f(x_k) + (1 - \mu)T_k + \gamma\alpha_k \langle d_k, g(x_k)\rangle,$$

$$T_k = \max\left[f(x_k), \sum_{r=0}^{m(k)-1} \lambda_{kr} f(x_{k-r})\right].$$

The following algorithm model is a modified nonmonotone memory gradient method in [14], here we replace the Armijo monotone line search by our modified nonmonotone line search technique. The direction d_k is determined by the following formula for solving problem (1):

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

$$d_k = \begin{cases} -g_k, & k = 1, \\ -g_k + \beta_k \delta_{k-1}, & k \geq 2, \end{cases} \tag{4}$$

where $\delta_k = d_{k-1} - g_{k-1}$ and the parameter β_k is chosen from Tang and Dong [14]

$$\beta_k = \begin{cases} 0, & \text{if } d_{k-1} = g_{k-1}, \\ \dfrac{\eta\| g_k \|}{\|\delta_{k-1}\|} & \text{if } d_{k-1} \neq g_{k-1}. \end{cases} \tag{5}$$

Algorithm 2.1.

Step 0. Give x_1, and $0 < \beta < 1$, $0 < \gamma < 1$, $\eta \in \left(\frac{1}{2}, 1\right)$, $\varepsilon > 0$, $k = 1$.

Step 1. Compute g_k, if $\|g_k\| \le \varepsilon$, stop.

Step 2. Compute d_k by (4) and (5).

Step 3. Let initial step $\alpha_k = 1$.

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

Step 5. Let M be a nonnegative integer, $\mu \in [0, 1]$. For each k, let $m(k)$ satisfy

$$m(0) = 0,\ 0 \le m(k) \le \min[m(k-1) + 1,\ M] \text{ for } k \ge 1,$$

$\alpha_k \ge 0$ is bounded above and satisfies

$$f(x_k + \alpha_k d_k) \le \mu f(x_k) + (1 - \mu) T_k + \gamma \alpha_k \langle d_k,\ g(x_k) \rangle, \tag{6}$$

$$T_k = \max\left[f(x_k),\ \sum_{r=0}^{m(k)-1} \lambda_{kr} f(x_{k-r})\right].$$

If (6) does not hold, define $\alpha_k = \alpha_k \beta$ and go to Step 4.

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

Note that the algorithm is well-defined, in fact, when $\alpha_k = 0$, $f(x_k + \alpha_k d_k) = f(x_k)$ there must exist a sufficient small α_k such that

$$f(x_k + \alpha_k d_k) \le f_k + \gamma \alpha_k \nabla f(x_k) d_k.$$

By $\nabla f(x_k) d_k \le 0$ and $0 < \gamma < 1$, together with

$$f(x_k) \le \mu f(x_k) + (1 - \mu) \max\left[f(x_k),\ \sum_{r=0}^{m(k)-1} \lambda_{kr} f(x_{k-r})\right].$$

Then we have

$$f(x_k + \alpha_k d_k) \le \mu f(x_k)$$

$$+ (1-\mu)\max\left[f(x_k),\ \sum_{r=0}^{m(k)-1} \lambda_{kr} f(x_{k-r}) \right] + \gamma\alpha_k \nabla f(x_k) d_k.$$

3. The Global Convergent Properties

In this section, we discuss the global convergence property of algorithm with the modified nonmonotone line search. In order to achieve the convergence of Algorithm 2.1, we give some Assumptions as follow:

Assumption 3.1.

A: $f(x)$ is bounded above on the level set $L = \{x | f(x) \le f(x_0)\}$.

B: In some neighborhood Ω of L, f is continuously differentiable, and its gradient $\nabla f(x)$ is Lipschitz continuous, namely, there exists a constant L such that

$$\|\nabla f(x) - \nabla f(x_k)\| \le L\|x - x_k\|.$$

Lemma 3.1. *Suppose Assumption* 3.1 *holds, and* d_k *is computed by* (4) *and* (5). *We have* $-g_k^T d_k \ge (1-\eta)\|g_k\|^2$, *for any* $k \ge 1$.

Proof. If $k = 1$, we have $-g_k^T d_k = \|g_k\|^2$, the conclusion holds.

If $k > 1$, we have

$$-g_k^T d_k = -g_k\left(-g_k + \frac{\eta\|g_k\|}{\|\eta_{k-1}\|}\eta_{k-1}\right),$$

$$-g_k^T d_k = \|g_k\|^2 \pm \eta\|g_k\|^2.$$

The conclusion follows.

Lemma 3.2. *Suppose Assumption* 3.1 *holds, and* d_k *is computed by* (4), (5). *We* *have* $\|d_k\| \leq (1+\eta)\|g_k\|$, *for any* $k \geq 1$.

Proof. If $k = 1$, we have $\|d_k\| = \|g_k\|$, holds.

If $k > 1$, we have

$$\|d_k\| = \left\| -g_k + \frac{\eta\|g_k\|}{\|\eta_{k-1}\|}\eta_{k-1} \right\|$$

$$\|d_k\| = \| -g_k \pm \eta g_k \|$$

holds.

Lemma 3.3. *Suppose Assumption B holds*, α_k *satisfy formula* (6) *of Algorithm* 2.1, *then there exists* β *such that*

$$\alpha_k \geq \min\left\{1, \frac{(1-\gamma)\beta}{L}\frac{|\langle g_k, d_k\rangle|}{\|d_k\|^2}\right\}. \tag{7}$$

Proof. At the kth iterate, if $\alpha_k = 1$ satisfies formula (6), then $\alpha_k = 1$. Otherwise, there exists β, which does not satisfy formula (6) for $\alpha_k/\beta > 0$, in other words, it holds,

$$f\left(x_k + \frac{\alpha_k}{\beta}d_k\right) > \mu f(x_k) + (1-\mu)T_k + \gamma\frac{\alpha_k}{\beta}\langle d_k, g(x_k)\rangle$$

$$> f(x_k) + \gamma\frac{\alpha_k}{\beta}\langle d_k, g(x_k)\rangle. \tag{8}$$

By mean value theorems, we have:

$$f(x_k + \alpha d_k) - f(x_k) = \int_0^\beta \langle g(x_k + td_k) - g(x_k)d_k\rangle d_t + \alpha\langle g(x_k), d_k\rangle$$

$$\leq \frac{1}{2}L\alpha^2\|d_k\|^2 + \alpha\langle g(x_k), d_k\rangle.$$

Together with (8), we have

$$\alpha_k \geq \min\left\{1, \frac{(1-\gamma)\beta}{L} \frac{|\langle g_k, d_k \rangle|}{\|d_k\|^2}\right\}.$$

Therefore, the conclusion holds.

Lemma 3.4. *Suppose Assumption* 3.1 *holds*, *and the sequence* x_k *is generated by Algorithm* 2.1, *then we have*

$$f(x_k) \leq f(x_0) + \lambda\gamma \sum_{r=0}^{k-2} \alpha_r \langle g(x_r), d_r \rangle + \gamma\alpha_{k-1} \langle g(x_{k-1}), d_{k-1} \rangle$$

$$\leq f(x_0) + \lambda\gamma \sum_{r=0}^{k-1} \alpha_r \langle g(x_r), d_r \rangle. \tag{9}$$

Proof. We prove the conclusion by induction.

If $k = 1$, by (6) and $\lambda \leq 1$, we have

$$f(x_1) \leq (x_0) + \lambda\alpha_0 \langle g(x_0), d_0 \rangle \leq f(x_0) + \gamma\lambda\alpha_0 \langle g(x_0), d_0 \rangle.$$

Assume (9) holds for $1, 2, \cdots, k$, we can consider by the following two cases:

Case 1. $\max\left[f(x_k), \sum_{r=0}^{m_k-1} \lambda_{kr} f(x_{k-r})\right] = f(x_k)$, by (6), we have

$$f(x_{k+1}) = f(x_k + \alpha_k d_k) \leq f(x_k) + \gamma\alpha_k \langle g(x_k), d_k \rangle$$

$$\leq f(x_0) + \lambda\gamma \sum_{r=0}^{k-1} \alpha_r \langle g(x_r), d_r \rangle + \gamma\alpha_k \langle g(x_k), d_k \rangle.$$

$$\leq f(x_0) + \lambda\gamma \sum_{r=0}^{k} \alpha_r \langle g(x_r), d_r \rangle.$$

Case 2. $\max\left[f(x_k), \sum_{r=0}^{m_k-1} \lambda_{kr} f(x_{k-r})\right] = \sum_{r=0}^{m_k-1} \lambda_{kr} f(x_{k-r})$. Let $q = \min[k, M-1]$, by (6) we have

$$f(x_{k+1}) = f(x_k + \alpha_k d_k)$$

$$\leq \mu f(x_k) + (1-\mu)\sum_{p=0}^{q} \lambda_{kp} f(x_{k-p}) + \gamma\alpha_k \langle g(x_k), d_k \rangle$$

$$\leq \mu f(x_k) + (1-\mu)\sum_{p=0}^{q} \lambda_{kp}[f(x_0) + \lambda\gamma \sum_{r=0}^{k-p-2} \alpha_r \langle g(x_r), d_r \rangle$$

$$+ \gamma\alpha_{k-p-1}\langle g(x_{k-p-1}), d_{k-p-1}\rangle] + \gamma\alpha_k \langle g(x_k), d_k \rangle.$$

By $(1, 2. \cdots, q) \times (1, 2, \cdots, k-q-2) \subset \{(p, r) : 0 \leq p \leq q, 0 \leq r \leq k-q-2\}$, $\sum_{p=0}^{q} \lambda_{kp} = 1, \lambda_{kp} \geq \lambda,$ we have

$$f(x_{k+1}) \leq \mu f(x_k) + (1-\mu)\left[f(x_0) + \lambda \sum_{r=0}^{k-q-2} \left(\sum_{p=0}^{q} \lambda_{kp} \right) \alpha_r \langle g(x_r), d_r \rangle \right.$$

$$\left. + \gamma \sum_{p=0}^{q} \lambda_{kp} \alpha_{k-p-1} \langle g(x_{k-p-1}), d_{k-p-1} \rangle \right] + \gamma\alpha_k \langle g(x_k), d_k \rangle$$

$$\leq \mu f(x_k) + (1-\mu)\left[f(x_0) + \lambda\gamma \sum_{r=0}^{k-q-2} \alpha_r \langle g(x_r), d_r \rangle \right.$$

$$\left. + \lambda\gamma \sum_{r=k-p-1}^{k-1} \alpha_r \langle g(x_r), d_r \rangle \right] + \gamma\alpha_k \langle g(x_k), d_k \rangle$$

$$= \mu f(x_k) + (1-\mu)\left[f(x_0) + \lambda\gamma \sum_{r=0}^{k-1} \alpha_r \langle g(x_r), d_r \rangle \right]$$

$$+ \gamma\alpha_k \langle g(x_k), d_k \rangle.$$

By the case 1, we have:

$$f(x_{k+1}) \leq \mu \left[f(x_0) + \lambda\gamma \sum_{r=0}^{k-1} \alpha_r \langle g(x_r), d_r \rangle \right]$$

$$+ (1-\mu)\left[f(x_0) + \lambda\gamma\sum_{r=0}^{k-1}\alpha_r\langle g(x_r), d_r\rangle\right] + \gamma\alpha_k\langle g(x_k), d_k\rangle$$

$$\leq f(x_0) + \lambda\gamma\sum_{r=0}^{k-1}\alpha_r\langle g(x_r), d_r\rangle + \gamma\alpha_k\langle g(x_k), d_k\rangle$$

$$\leq f(x_0) + \lambda\gamma\sum_{r=0}^{k}\alpha_r\langle g(x_r), d_r\rangle.$$

The conclusion follows.

Theorem 3.1. *Suppose Assumption* 3.1 *holds*, *and the sequences* x_k *and* d_k *are generated by Algorithm* 2.1, *then*

$$\lim_{k\to\infty}\langle g(x_k), d_k\rangle = 0. \tag{10}$$

Proof. To get a contradiction, we assume that there exist a boundless sequence index set K, and $\varepsilon > 0$, which satisfy $\langle g(x_k), d_k\rangle \leq -\varepsilon$ for any $k \in K$, based on Lemma 3.4, for any $k \in K$, we have,

$$-\lambda\gamma\sum_{r=0,\, r\in K}^{k-1}\alpha_r\langle g(x_r), d_r\rangle \leq f(x_0) - f(x_k). \tag{11}$$

By Lemma 3.1, $-g_k^T d_k \geq (1-\eta)\|g_k\|^2$, it means

$$-\frac{\langle g(x_k), d_k\rangle}{\|g_k\|^2} \geq (1-\eta). \tag{12}$$

Together with (7), (11) and (12), we have:

$$f(x_0) - f(x_k) \geq -\lambda\gamma\sum_{r=0,\, r\in K}^{k-1}\alpha_r\langle g(x_r), d_r\rangle$$

$$\geq \lambda\gamma\varepsilon\sum_{r=0,\, r\in K}^{k-1}\alpha_r$$

$$\geq \lambda\gamma\varepsilon \sum_{r=0,\, r\in K}^{k-1} \min\left\{1,\ \frac{(1-\gamma)\beta}{L} \cdot \frac{\langle g(x_r),\ d_r\rangle}{\|d_r\|^2}\right\}$$

$$\geq \lambda\gamma\varepsilon \sum_{r=0,\, r\in K}^{k-1} \min\left\{1,\ \frac{(1-\gamma)\beta}{L} \cdot (1-\eta)\right\}.$$

Since $f(x)$ is bounded below, let $k \to \infty$ $(k \in K)$, we have

$$\infty \geq f(x_0) - f(x_k) \to \infty$$

which is a contradiction, so the conclusion holds.

Theorem 3.2. *Suppose Assumption* 3.1 *holds, and the sequence* $\{x_k\}$ *is generated by Algorithm* 2.1, *then we have*

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

Proof. By Lemma 3.1 and Theorem 3.1, we have

$$0 \geq \lim_{k\to\infty} (\eta - 1)\|g_k\|^2 \geq \lim_{k\to\infty} \langle g(x_k),\ d_k\rangle = 0$$

which implies

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

4. Numerical Examples

In this section, we give the numerical results that are obtained from the standard test problems. We fixed the value of μ_k, which seems to work reasonably well for a broad class of problems. Let $\eta = 0.88,\ \beta = 0.5,\ M = 10,\ \gamma = 0.75,\ \lambda_{kr} = \frac{1}{m(k)}$, $(r = 0, 1, \cdots, m(k) - 1)$, the termination criterion is $\|g_k\| \leq 10^{-5}$. All the problems are computed in Matlab 7.1.

Problem 1. $f(x) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2$,

$$x_0 = [-1.2, 1]', \ x^* = [1, 1], \ f(x^*) = 0.$$

Problem 2. $f(x) = 100(x_1^2 - x_2)^2 + (x_1 - 1)^2 + (x_3 - 1)^2 + 90(x_3^2 - x_4)^2$

$+ 10.1[(x_2 - 1)^2 + (x_4 - 1)^2] + 19.8(x_2 - 1)(x_4 - 1),$

$$x_0 = [-3, -1, -3, -1]', \ x^* = [1, 1, 1, 1], \ f(x^*) = 0.$$

Problem 3. $f(x) = (x_1 + 10x_2)^2 + 5(x_3 - x_4)^2 + (x_2 - 2x_3)^4 + 10(x_1 - x_4)^4,$

$$x_0 = [3, -1, 0, 1]', \ x^* = [0, 0, 0, 0], \ f(x^*) = 0.$$

Problem 4. $f(x) = 100(x_2 - x_1^3)^2 + (1 - x_1)^2,$

$$x_0 = [-1.2, -1]', \ x^* = [1, 1], \ f(x^*) = 0.$$

Problem 5. $f(x) = (x_1 + 10 * x_2)^4 + 5 * (x_3 - x_4)^4 + (x_2 - 2 * x_3)^4 + 10 * (x_1 - 10 * x_4)^4,$

$$x_0 = (2, 2, -2, -2)^T, \ x^* = (0, 0, 0, 0)^T, \ f^* = 0.$$

Problem 6. $f(x) = (x_1 - 1)^2 + (x_1 - x_2)^2 + (x_3 - 1)^2 + (x_4 - 1)^4 + (x_5 - 1)^6,$

$$x_0 = (2, 2, 2, 2, 2)^T, \ x^* = (1, 1, 1, 1, 1)^T, \ f^* = 0.$$

Table 1.

Algorithm 1	$\mu = 0$	$\mu = 0.1$	$\mu = 0.2$	$\mu = 0.3$	$\mu = 0.4$	$\mu = 0.5$
Problem 1	288	271	467	546	677	577
Problem 2	4303	4223	4468	4690	4333	3815
Problem 3	338	672	734	99	1122	872
Problem 4	1796	1587	1349	1772	1958	1341

Problem 5	493	495	179	137	177	152
Problem 6	1124	1187	1001	923	733	729

Algorithm 1	$\mu = 0.6$	$\mu = 0.7$	$\mu = 0.8$	$\mu = 0.9$	$\mu = 1$
Problem 1	535	673	644	617	943
Problem 2	4126	3836	3954	3850	4282
Problem 3	405	1020	1168	1176	4326
Problem 4	1519	1305	1049	1479	2732
Problem 5	336	349	293	170	654
Problem 6	717	101	1170	1285	1762

Given $M = 10$ in Algorithm 2.1, the results in Table 1 indicate that the numerical results depend on the value of the parameter μ. It is easy to see that the modified nonmonotone line search is actually the monotone Armijo line search when $\mu = 1$, and the nonmonotone line search proposed by Yu and Pu [15] when $\mu = 0$. We can see the numerical results are efficient and improved in most examples.

Moreover, we also compare our method with other methods proposed in [14] and [23], the results are listed in Table 2, where $\mu = 0.1$, $M = 10$ are given in Algorithm 2.1, $M = 10$ in [23] and n_l represents the number of successful iterations.

Table 2.

Problem 1	Algorithm in [14]	Algorithm in [23]	Algorithm 2.1
n_l	943	288	271
Problem 2	Algorithm in [14]	Algorithm in [23]	Algorithm 2.1
n_l	4282	4303	3815

Problem 3	Algorithm in [14]	Algorithm in [23]	Algorithm 2.1
n_l	4326	338	99
Problem 4	Algorithm in [14]	Algorithm in [23]	Algorithm 2.1
n_l	2732	1796	1049
Problem 5	Algorithm in [14]	Algorithm in [23]	Algorithm 2.1
n_l	654	493	137
Problem 6	Algorithm in [14]	Algorithm in [23]	Algorithm 2.1
n_l	1762	1124	101

References

[1] J. C. Gilbert and J. Nocedal, Global convergence properties of conjugate gradient methods for optimization, SIAM J. Optim. 2 (1992), 21-42.

[2] A. Miele and J. W. Cantrell, Study on a memory gradient method for the minmization of functions, J. Optim. Theory Appl. 3 (1969), 459-470.

[3] L. Grippo and S. Lucidi, A globally convergent version of the Polak & Ribiere conjugate gradient method, Math. Prog. 78 (1997), 375-391.

[4] J. Nocedal, Theory of algorithm for unconstrained optimization, Acta Numerica, Cambridge University Press, 1992.

[5] Z. J. Shi, Restricted PR conjugate gradient method and its global convergence, Adv. Math. 31 (2002), 47-55.

[6] E. E. Cragg and A. V. Levy, Study on Supermemory gradient method for the minimization of function, J. Optim. Theory Appl. 4 (1969), 191-205.

[7] M. A. Wolfe and C. Viazminsky, Supermemory descent methods for unconstrained minimization, J. Optim. Theory Appl. 18 (1976), 455-468.

[8] Z. Shi, A new memory gradient under exact line search, Asia-Pacific J. Oper. Res. 20 (2003), 275-284.

[9] Z. Shi, A supermemory gradient method for unconstrained optimization, J. Eng. Math. 17 (2000), 99-104.

[10] W. Cantrell, Relation between the memory gradient method and the Fletcher-Revees method, J. Optim. Theory Appl. 4 (1969), 67-71.

[11] Y. Ou and G. Wang, A new supermemory gradient method for unconstrained optimization problems, Optim. Lett. 6 (2012), 975-992.

[12] D. C. Liu and J. Nocedal, On the limited memory BFGS method for large scale optimization, Math. Prog. 45 (1989), 503-528.

[13] J. Nocedal, Updating quasi-Newton matrices with limited storage, Math. Comput. 35 (1980), 773-782.

[14] J. Tang and L. Dong, A new memory gradient method and its convergence, Gongcheng Shuxue Xuebao 27 (2010), 636-642.

[15] Z. Yu and D. Pu, A new nonmonotone line search technique for unconstrained optimization, Comput. Appl. Math. 219 (2008), 134-144.

[16] H. Zhang and W. Hager, A nonmonotone line search technique and its application to unconstrained optimization, SIAM J. Optim. 14 (2004), 1043-1056.

[17] L. Grippo, F. Lampariello and S. Ludidi, A nonmonotone line search technique for Newton's method, SIAM J. Numerical Anal. 23 (1986), 707-716.

[18] R. M. Chamberlain, M. J. D. Powell, C. Lemarechal and H. C. Pedersen, The watchdog technique for forcing convergence in algorithm for constrained optimization, Math. Program. Stud. 16 (1982), 1-17.

[19] P. L. Toint, An assessment of nonmonotone line search techniques for unconstrained optimization, SIAM J. Sci. Comput. 17 (1996), 725-739.

[20] Y. H. Dai, On the nonmonotone line search, J. Optim. Theory Appl. 112 (2002), 315-330.

[21] Z. Shi and J. Shen, Convergence of nonmonotone line search method, J. Comput. Appl. Math. 193 (2006), 397-412.

[22] M. Raydan, The Barzilai and Borwein gradient method for large scale unconstrained minimization problem, SIAM J. Optim. 7 (1996), 26-33.

[23] Z. X. Rong and K. Su, A new nonmonotone memory gradient method for unconstrained optimization, Math. Aeterna 5 (2015), 635-647.