Mathematica Aeterna, Vol. 5, 2015, no. 4, 635 - 647

A New Nonmonotone Memory Gradient Method for Unconstrained Optimization

Zixing Rong

College of Mathematics and information Science, Hebei University, Baoding, 071002 China

Ke Su

College of Mathematics and information Science, Hebei University, Baoding, 071002 China

Abstract

In this paper, we introduced a new nonmonotone line search technique, and then we proposed a modified nonmonotone memory gradient method, where we replace the Armijo monotone line search by the new nonmonotone line search technique. theoretical analysis and numerical results indicate our algorithm have some advantages. what is more, we prove the global convergence of the algorithm, in the end numerical results obtained for a set of standard test problems are reported which indicate that the proposed methods is highly effective if we choose a good parameter in the method.

Mathematics Subject Classification: xxxxx

Keywords: unconstrained optimization, memory gradient methods, global convergence, modified nonmonotone line search

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 an Euclidean space. many iterative methods for (1) produce a sequence x_0, x_1, x_2, $\cdots$, where x_{k+1} is generated from x_k, the current direction d_k, and the stepsize $\alpha_k > 0$ by the rule

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

Unconstrained optimization problem is an important research topic in mathematical programming fields. There exist many methods for solving unconstrained optimization problem, these methods have research topic in many fields such as economy, management, control science, etc. It is well-known that 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. The search direction d_k conjugate gradient methods is recursively defined by

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

where β_k is a parameter which determines different conjugate gradient methods. The best-known formulas for β_k are Fletcher-Reeves (FR), Polark-Ribire-Polyak (PRP), Polark-Ribire-Polyak plus (PRP+) and Dai-Yuan (DY), etc. Moreover, the global convergence properties of conjugate gradient methods have been studied by many researchers [3,4,5].

The search direction d_k in memory gradient method does not have a uniform form. Compared with conjugate gradient method, memory gradient method can use the information of the previous iterations more sufficiently and hence it is helpful to design algorithms with quick convergence rate. In line search method, the search direction d_k is generally required to satisfy

$$g_k^T d_k < 0$$

which guarantees that d_k is a descent direction of $f(x)$ at x_k. In order to guarantee the global convergence, we sometimes require d_k to satisfy a sufficient descent condition.

The first idea of memory gradient methods was proposed by Miele and Cantrell (1969)[2] and Cragg and Levy (1969)[6]. Memory and super-memory gradient methods are similar to conjugate gradient methods. Comparing with conjugate gradient methods, however, memory gradient methods sufficiently use the previous multi-step iterative information at every iteration. Moreover, they add the freedom of some parameters and enable us to design some quick convergent and robust method. Many authors have studied the global convergence properties for these methods and yielded substantial results [7-11]. Some memory gradient methods have been proposed and investigated in the literature [1,12,13], memory gradient method is a good method for solving the unconstrained optimization problem in management and engineering.

In this paper, we proposed a modified nonmonotone memory gradient method in [14], here we replace the Armijo monotone line search in Step 5 by the new 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. firstly we introduced a new mon-monotone line search technique by Yu [15] and then we combined a memory gradient method [14] with the new nonmonotone line search technique. which perform efficiently both in theory and numerical results than the algorithm NA in [14]. The direction generated by the new method automatically satisfies sufficient descent condition at every iteration without requiring conditions, and we given the proof in the lemma 1.1. 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.

The paper is organized as follows. in section 2, we give a new nonmonotone line search proposed by Yu [15], and then we combined a memory gradient method with the new nonmonotone line search technique. what is more, we give the algorithm. in section 3, we establish the global convergence of the algorithm, in section 4, all the essential parameters of its implementation is given, some numerical experiments are given.

2 Algorithm

After the direction d_k is determined the next task is to find a step size α_k along the search direction. The ideal line search rule is the exact one which satisfies:

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

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, Goldtein rule, Wolfe rule and nonmontone line search [17]

In 1982, Chamberlain et al. in [18] proposed a watchdog technique for constrained optimization, in which some standard line search conditions were relaxed to overcome the Marotos effect. Motivated by this idea, Grippo, Lampariello and Lucidi in [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 nonmontone 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, Lampariello, and Lucidi in [17] for Newton′s methods. thanks to its excellent numerical exhibition, many nonmonotone techniques have been developed in recent years, before introduce the new nonmonotone technique, we describe the nonmonotone Armijo rule. α_k is a stepsize with $\alpha_k \geq 0$ and d_k is a search direction satisfied $g_k^T d_k \leq 0$, Let $a > 0$, $\gamma \in (0,1)$, $\beta \in (0,1)$and let M be a nonnegative integer. For each k , let $m(k)$ satisfies

$$m(0) = 0, 0 \leq m(k) \leq \min[m(k-1)+1, M], for\ k \geq 1,$$

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, so α_k should satisfied 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, so α_k should satisfied 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 $0 < \gamma_1 \leq \gamma_2 < 1$

the nonmonotone line search methods have been studied by many authors Toint (1996); Dai (2002); Zhang and Hager (2004); Shi and Shen (2006); Yu and Pu (2008); Hadi Nosratipour (2013). Theoretical analysis and numerical results show that the nonmonotone algorithms are very efficient.

Although these nonmonotone technique work well in many case, there are some drawbacks, First, a good function value generated in any iteration is essentially discard due to the max. Second, in some case, the numerical performance is very dependent on the choice of M [17.19.20] now we give a new nonmonotone line search proposed by Yu as follows: [15]

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

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

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

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

The following algorithm model is a modified nonmonotone memory gradient method in [14], here we replace the Armijo monotone line search in Step 5 by the new nonmonotone line search technique by Yu [15]. the direction d_k is determined by the following formula for solving problem (1):

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

$$d_k = \begin{cases} -g_k, & k = 1, \\ -g_k + \beta_k \delta_{k-1}, & k \geq 2. \end{cases} \tag{3}$$

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

$$\beta_k = \begin{cases} 0, & \text{if } d_{k-1} = g_{k-1}, \\ \frac{\eta \|g_k\|}{\|\delta_{k-1}\|}, & \text{if } d_{k-1} \neq g_{k-1}. \end{cases} \tag{4}$$

Algorithm 2.1

Step0. Give starting guess x_1, and some constants, $0 < \beta < 1$, $0 < \delta < 1$, $\eta \in (\frac{1}{2}, 1)$, $\varepsilon > 0$ and $k = 1$.

Step1. Compute g_k, and $\|g_k\| \leq \varepsilon$, STOP.

Step2. Compute d_k by (3) (4).

Step3. Set trial step $\alpha_k = 1$.

Step4. Set $x_{k+1} = x_k + \alpha_k d_k$.

Step5. Let $\lambda \in (0, 1]$ $M \geq 1$ is a positive integer,defined $m(k) = \min[k + 1, M]$

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

Let $\alpha_k \geq 0$ be bounded above and satisfy:

$$f(x_k + \alpha_k d_k) \leq \max[f(x_k), \sum_{r=0}^{m(k)-1} \lambda_{kr} f(x_{k-r})] + \gamma \alpha_k \langle d_k, g(x_k) \rangle. \tag{5}$$

If (6) does not holds, define $\alpha_k = \alpha_k \beta$ and go to step 4

Step6. Set $k := k + 1$, and go to step 1 .

Note that the algorithm cannot cycle indefinitely in step 5 in fact, when $\alpha_k = 0$, $f(x_k + \alpha_k d_k) = f(x_k)$ there must exist a sufficient small α_k

$$f(x_k + \alpha_k d_k) \leq f_k + \gamma \alpha_k \nabla f(x_k) d_k,$$

because of $\nabla f(x_k) d_k \leq 0$ and $0 < \gamma < 1$. what is more

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

So we have

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

3 The Global Convergence of the Algorithm

In this section, we discuss the global convergence property of algorithm with the nonmonotone line search technique by Yu[15]. In order to achieve the convergence of the second algorithm 2.1, we give some Assumptions as follow:
Assumption 3.1

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

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

$$\|\nabla f(x) - \nabla f(x_k)\| \leq L \|x - x_k\|$$

Lemma 3.1 d_k is computed by (3)(4), We have $-g_k^T d_k \geq (1-\eta)\|g_k\|^2$, for any $k \geq 1$
Proof: if $k = 1$ we have $-g_k^T d_k = \|g_k\|^2$, holds
if $k > 1$ we have

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

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

holds end.
Lemma 3.2 d_k is computed by (3)(4), 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

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

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

holds end.
Lemma 3.3 Assumption B is hold, α_k satisfy the formula(5) of algorithm, then there exist β satisfy :

$$\alpha_k \geq \min\{1, \frac{(1-\gamma)\beta}{L} \frac{|\langle g_k, d_k \rangle|}{\|d_k\|^2}\}. \quad (6)$$

Proof:

At the kth iterate, if $\alpha_k = 1$ satisfy the formula (5). then $\alpha_k = 1$,

otherwise, there exist β , which does not satisfy formula (5) for $\alpha_k/\beta > 0$, in other words :

$$\begin{aligned} f(x_k + \frac{\alpha_k}{\beta}d_k) &> \mu f(x_k) + (1-\mu)T_k + \gamma\frac{\alpha_k}{\beta}\langle d_k, g(x_k)\rangle \\ &> f(x_k) + \gamma\frac{\alpha_k}{\beta}\langle d_k, g(x_k)\rangle. \end{aligned} \tag{7}$$

by mean value theorems, we have:

$$\begin{aligned} f(x_k + \alpha d_k) - f(x_k) &= \int_0^\beta \langle g(x_k + td_k) - g(x_k)d_k\rangle d_t + \alpha\langle g(x_k), d_k\rangle. \\ &\leq \frac{1}{2}L\alpha^2\|d_k\|^2 + \alpha\langle g(x_k), d_k\rangle. \end{aligned}$$

with(7), we have

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

So, (6)is hold.

Lemma 3.4 Assume that sequence x_k is generated by algorithm 2.1, so that:

$$\begin{aligned} f(x_k) &\leq f(x_0) + \lambda\gamma\sum_{r=0}^{k-2}\alpha_r\langle g(x_r), d_r\rangle + \gamma\alpha_{k-1}\langle g(x_{k-1}), d_{k-1}\rangle \\ &\leq f(x_0) + \lambda\gamma\sum_{r=0}^{k-1}\alpha_r\langle g(x_r), d_r\rangle. \end{aligned} \tag{8}$$

Proof: We prove by induction.

if $k = 1$, by (5) and $\lambda \leq 1$, we have

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

Assume (8) is hold for 1 , 2 $\cdots$ k , we can think of this problem from two cases :

case 1 : $\max[f(x_k), \sum_{r=0}^{m_k-1}\lambda_{kr}f(x_{k-r})] = f(x_k)$, by (5),we have

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

case 2 : $\max[f(x_k), \sum_{r=0}^{m_k-1} \lambda_{kr} f(x_{k-r})] = \sum_{r=0}^{m_k-1} \lambda_{kr} f(x_{k-r})$,Let$q = \min[k, M-1]$ by (5) we have

$$
\begin{aligned}
f(x_{k+1}) = f(x_k + \alpha_k d_k) &\leq \sum_{p=0}^{q} \lambda_{kp} f(x_{k-p)}) + \gamma\alpha_k \langle g(x_k), d_k\rangle. \\
&\leq \sum_{p=0}^{q} \lambda_{kp}[f(x_0) + \lambda\gamma \sum_{r=0}^{k-p-2} \alpha_r \langle g(x_r), d_r\rangle + \\
&\gamma\alpha_{k-p-1}\langle g(x_{k-p-1}), d_{k-p-1}\rangle] + \gamma\alpha_k\langle g(x_k), d_k\rangle.
\end{aligned}
$$

impose (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

$$
\begin{aligned}
f(x_{k+1}) &\leq f(x_0) + \lambda \sum_{r=0}^{k-q-2} (\sum_{p=0}^{q} \lambda_{kp})\alpha_r\langle g(x_r), d_r\rangle \\
&+ \gamma \sum_{p=0}^{q} \lambda_{kp}\alpha_{k-p-1}\langle g(x_{k-p-1}), d_{k-p-1}\rangle + \gamma\alpha_k\langle g(x_k), d_k\rangle. \\
&\leq f(x_0) + \lambda\gamma \sum_{r=0}^{k-q-2} \alpha_r\langle g(x_r), d_r\rangle + \lambda\gamma \sum_{r=k-p-1}^{k-1} \alpha_r\langle g(x_r), d_r\rangle \\
&+ \gamma\alpha_k\langle g(x_k), d_k\rangle \\
&= f(x_0) + \lambda\gamma \sum_{r=0}^{k-1} \alpha_r\langle g(x_r), d_r\rangle + \gamma\alpha_k\langle g(x_k), d_k\rangle \\
&\leq f(x_0) + \lambda\gamma \sum_{r=0}^{k} \alpha_r\langle g(x_r), d_r\rangle
\end{aligned}
$$

end.

Theorem 3.1 Assume that x_k and d_k is generated by algorithm 2.1 and A, B holds, and then

$$\lim_{k\to\infty} \langle g(x_k), d_k\rangle = 0. \tag{9}$$

Proof:

Assume that there exist a boundless sequence index set K, and there exist $\varepsilon > 0$, which satisfy $\langle g(x_k), d_k\rangle \leq -\varepsilon$. for any $k \in K$, impose the lemma 3.4 , for any $k \in K$, we have :

$$-\lambda\gamma \sum_{r=0 r\in K}^{k-1} \alpha_r\langle g(x_r), d_r\rangle \leq f(x_0) - f(x_k), \tag{10}$$

impose the lemma 3.1 , $-g_k^T d_k \geq (1-\eta)\|g_k\|^2$, which means

$$-\frac{\langle g(x_k), d_k\rangle}{\|g_k\|^2} \geq (1-\eta), \tag{11}$$

with (6) (10), (11) we have:

$$\begin{aligned} f(x_0) - f(x_k) &\geq -\lambda\gamma \sum_{r=0, r\in K}^{k-1} \alpha_r \langle g(x_r), d_r\rangle \\ &\geq \lambda\gamma\varepsilon \sum_{r=0, r\in K}^{k-1} \alpha_r \\ &\geq \lambda\gamma\varepsilon \sum_{r=0, r\in K}^{k-1} \min\{1, \frac{(1-\gamma)\beta}{L} \cdot \frac{|\langle g(x_r), d_r\rangle|}{\|d_r\|^2}\} \\ &\geq \lambda\gamma\varepsilon \sum_{r=0, r\in K}^{k-1} \min\{1, \frac{(1-\gamma)\beta}{L} \cdot (1-\eta)\} \end{aligned}$$

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

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

paradoxically, so (9) holds
end.
Theorem 3.2 Assume that sequence $\{x_k\}$ is generated by algorithm 2.1, and 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

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

end.

4 Numerical Examples

In this section, we report the numerical results obtained for a set of standard tests problems. to test the effect of the algorithms, we selected a few examples of numerical experiments, Although our best convergence result of the first were obtained by dynamically varying μ_k, in this numerical test we take some fixed value μ_k, which seemed to work reasonably well for a broad class of problems. in the two algorithms, $\eta = 0.88$, $\beta = 0.5$, $M = 10$, $\gamma = 0.75$, $\lambda_{kr} = \frac{1}{m(k)}, (r = 0, 1, \cdots, m(k) - 1)$, For all codes the termination criterion was $\|g_k\| \leq 10^{-5}$ except problem 3 whose termination criterion was $\|g_k\| \leq 10^{-4}$, we compute the number n_l of line search.

The algorithms has been tested on the following set of problems by MATLAB 7.1.

$Problem1.f(x) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2$

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

$Problem2.f(x) = 100(x_1^2 - x_2)^2 + (x_1 - 1)^2 + (x_3 - 1)^2 + 90(x_3^2 - x_4)^2 + 10.1[(x_2 - 1)^2 + (x_4 - 1)^2] + 19.8(x_2 - 1)(x_4 - 1)$

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

$Problem3.f(x) = (x_1 + 10x_2)^2 + 5(x_3 - x_4)^2 + (x_2 - 2x_3)^4 + 10(x_1 - x_4)^4$

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

$Problem4.f(x) = 100(x_2 - x_1^3)^2 + (1 - x_1)^2$

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

$Problem5.f(x) = (x_1 + 10 * x_2)^4 + 5 * (x_3 - x_4)^4 + (x_2 - 2 * x_3)^4 + 10 * (x_1 - 10 * x_4)^4$

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

$Problem6.f(x) = (x_1 - 1)^2 + (x_1 - x_2)^2 + (x_3 - 1)^2 + (x_4 - 1)^4 + (x_5 - 1)^6$

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

TABLE 1

*results for algorithm*2.1

Algorithm 1	problem 1	pro 2	pro 3	pro 4	pro 5	problem 6
M=1	943	4282	4326	2732	654	1762
M=2	681	2972	1624	1270	408	1286
M=3	610	3601	1172	1117	232	471
M=4	510	3717	1112	1179	656	654
M=5	612	3749	1111	1675	543	724
M=6	633	3917	1088	1679	194	1000
M=7	625	3856	1216	1682	159	1218
M=8	628	3990	599	1572	145	1125
M=9	324	4061	957	1686	317	1068
M=10	288	4303	338	1796	493	1124

In algorithm 2.1, the numerical performance is highly dependent on the choice of M. almost the choice of M is better than the choice of $M = 1$ which is monotone Armijo line search in algorithm NA [14].

TABLE 2

results of the comparison between different methods

problem 1	NA	$algorithm 2.1(M = 10)$
n_l	943	288
problem 2	NA	$algorithm 2.1(M = 2)$
n_l	4282	2972
problem 3	NA	$algorithm 2.1(M = 10)$
n_l	4326	338
problem 4	NA	$algorithm 2.1(M = 3)$
n_l	2732	1117
problem 5	NA	$algorithm 2.1(M = 7)$
n_l	654	159
problem 6	NA	$algorithm 2.1(M = 3)$
n_l	1762	471

In the end, the algorithm NA [14] was compared with algorithm 2.1. the excellent numerical performance in table 2 indicates that the algorithm 2.1 is highly efficient than the algorithm NA in [14] if we choose appropriate parameters included in the method, besides theoretical analysis suggests the algorithms have some advantages.

5 Acknowledgements

This work is supported by the National Natural Science Foundation of China (No. 11101115), the Natural Science Foundation of Hebei Province (No. 2014201033), the Key Research Foundation of Education Bureau of Hebei Province, (No.ZD2015069) and the Science and Technology project of Hebei province (No. 13214715).

References

[1] J.C. Gilbert, J. Nocedal. Global convergence properties of conjugate gradient methods for optimization, SIAM J. Optim. 2(1) 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(6) 1969, 459-470.

[3] L. Grippo, S. Lucidi. A globally convergent version of the PolakCRibie re conjugate gradient method, Math. Prog. 781997, 375-391.

[4] J. Nocedal. Theory of algorithm for unconstrained optimization, Acta Numerica, Cambridge University Press1992.

[5] Z.J. Shi. Restricted PR conjugate gradient method and its global convergence, Adv. Math. 31 (1)2002, 47-55.

[6] E.E.Cragg and A.V.Levy. study on Supermemory gradient method for the minimization of function, J. Optim. Theory and Applications,41969, 191-205.

[7] Wolfe, M.A., and Viazminsky, C. Supermemory descent methods for unconstrained minimization, J. Optim. Theor. Appl. 18(4),1976, 455-468.

[8] Shi, Z. A new memory gradient under exact line search, Asia-Pacific J. Oper. Res. 20(2), 2003, 275-284.

[9] Shi, Z. Hager. Supermemory gradient method for unconstrained optimization, J. Eng. Math. 17(2), 2000, 99-104.

[10] Cantrell, W. Relation between the memory gradient method and the Fletcher-Revees method, J.Optim. Theor. Appl. 4(1), 1969, 67-71.

[11] Ou, Y., Wang, G. A new supermemory gradient method for unconstrained optimization problems, Optim. Lett, 2011, doi:10.1007/s11590-011-0328-9.

[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] Tang, JingYong, and Dong, Li. A new memory gradient method and its convergence, Gongcheng Shuxue Xuebao Vol 27, 2010, 636-642.

[15] Zhensheng Yu and Dingguo Pu. A new nonmonotone line search technique for unconstrained optimization, Computational and Applied Mathematics, 219 2008, 134-144.

[16] Hongchao Zhang and William 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. Numer. Anal. 23, 1986, 707-716.

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

[19] M. Raydan, The Barzilai and Borwein gradient method for large scale unconstrained minimization problem, SIAM J. Optim.,71996, 26-33

[20] P. L. Toint, An assessment of nonmonotone line search techniques for unconstrained optimization, SIAM J. Sci. Comput.,171996, 725-739.

Received: August, 2015