JMathematica Aeterna, Vol. 5, 2015, no. 4, 615 - 627

A Memory Gradient Method under new Nonmonotone Line Search Technique for Unconstrained Optimization

Ke Su

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

Zixing Rong

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

Abstract

In this paper, we combine a new Memory gradient methods with a nonmonotone line search technique and we obtain a new algorithm. which perform efficiently both in theoretical and numerical results, what is more, we prove the global convergence of the algorithm under mild conditions, in the end Our numerical results show that the proposed method is efficient for given standard test problems if we choose a good parameter included in the method, furthermore, there are some advantages In theory.

Mathematics Subject Classification: xxxxx

Keywords: unconstrained optimization, memory gradient methods, global convergence, 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, most of methods for solving problem (1) are iterative methods which take the form

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

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 memory gradient method [1] is a good method for solving the unconstrained optimization problem in management and engineering. As is well known, conjugate gradient methods [2] are very important for solving problem (1). In particular, they are available for solving large scale unconstrained optimization problems because their storage and computation of some matrices associated with the Hessian of objective functions are relatively small. Therefore, the study of these methods attracted interest of many researchers. In general, the search direction of conjugate gradient methods can be written as:

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

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].

Memory and super-memory gradient methods are similar to conjugate gradient methods. The first idea of memory gradient methods was proposed by Miele and Cantrell (1969)[1] and Cragg and Levy (1969) [6]. Not only is their basic idea simple but also they avoid computation and storage of some matrices so that they are suitable to solve large scale unconstrained optimization problems. 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 [7-9]. Some memory gradient methods have been proposed and investigated in the literature [1,9,11]. However, the results on global convergence of memory gradient methods are rarely seen in the recent literature and need to be investigated thoroughly

Recently, *Tang* and *Dong* [12] proposed a memory gradient method which was determined as follows:

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

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

where $\delta_k = d_{k-1} - g_{k-1}$ and the parameter β_k is chosen from *Tang* and *Dong* [12]

In this paper, we proposed a modified nonmonotone memory gradient method based on [12], here we replace the Armijo monotone line search in Step

5 by a new nonmonotone line search technique [13]. the stepsize determined by Armijo monotone line search may considerably slow the rate of convergence in the presence of the narrow curved valley. we used the nonmonotone strategy to design the memory gradient methods, which perform efficiently both in theory and numerical results. 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 and a modified algorithm is proposed, all the essential features of its implementation is given, in section 3, we establish the global convergence of the algorithm under mild conditions, in section 4, 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 [14]

In 1982, Chamberlain et al. in [15] 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 [14] 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 [14] for Newton$'$s methods. 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

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

If the search is the nonmonotone Goldstein line search, so α_k should satisfied the following condition:

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

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

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:

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

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

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 [14.16.17] now we give a new nonmonotone line search proposed by Zhang and W.Hager as follows: [13]

Initialization : Choose starting guess x_0, and parameter $0 \leq \eta_{min} \leq \eta_{max} \leq 1$, $0 < \delta < 1 < \rho$ and $\mu > 0$. Set $C_0 = f(x_0)$, $Q_0 = 1$, and $k = 0$.

Convergence test : If $\|\nabla f(x_k)\|$ sufficiently small, then stop.

Line search update : Set $x_{k+1} = x_k + \alpha_k d_k$ Where α_k satisfies the nonmonotone Armijo condition:$\alpha_k = \bar{\alpha}_k \rho^{h_k}$.

$$f(x_k + \alpha_k d_k) \leq C_k + \delta \alpha_k \nabla f(x_k) d_k, \tag{3}$$

Where $\bar{\alpha}_k > 0$ is the trial step, and h_k is the largest integer such that (4) holds and $\alpha_k \leq \mu$.

Cost update : Choose $\eta_k \in [\eta_{min}, \eta_{max}]$, and set

$$Q_{k+1} = \eta_k Q_k + 1, C_{k+1} = (\eta_k Q_k C_k + f(k+1))/Q_{k+1} \tag{4}$$

Replace k by $k+1$ and return to the convergence test.

Observe that C_k is a convex combination of C_k and $f(x_{k+1})$.Since $C_0 = f(x_0)$,it follows that C_k is a convex combination of the function values $f(x_0)$, $f(x_1)$,$f(x_2)$,$\cdots$,$f(x_k)$.The choice of η_k controls the degree of nonmonotonicity. If $\eta_k = 0$ for each k, then the line is the usual monotone Armijo line search. If $\eta_k = 1$ for each k, then $C_k = A_k$, where

$$A_k = \frac{1}{k+1}\sum_{i=0}^{k} f_i, f_i = f(x_i),$$

is the average function value. the scheme with $C_k = A_k$ was suggested to us by Yu-hong Dai. In [18], the possibility of comparing the current function value with an average was analyzed.

We now present the new algorithm combining a memory gradient method with the nonmonotine line search, the following algorithm model is a modified nonmonotone memory gradient method based on [12], here we replace the Armijo monotone line search in Step 5 by the new nonmonotone line search technique [13]. the parameter β_k of the direction d_k is chosen from *Tang* and *Dong* [12] as following:

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

Algorithm 2.1

Step0. Give starting guess x_1, and some constants, $0 \leq \eta_{min} \leq \eta_{max} \leq 1$,$0 < \beta < 1$, $0 < \delta < 1$, $\eta \in (\frac{1}{2}, 1)$ Set $C_1 = f(x_1)$, $Q_1 = 1$, and $k = 1$.

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

Step2. Compute d_k by (2) (5).

Step3 Set trial step $\alpha_k = 1$.

Step4 Line search update:

$$f(x_k + \alpha_k d_k) \leq C_k + \delta\alpha_k \nabla f(x_k) d_k, \tag{6}$$

Where α_k satisfies the nonmonotone Armijo condition: $\alpha_k = \alpha_k \beta^{h_k}$. h_k is the smallest integer such that (6) holds.

Cost update: Choose $\eta_k \in [\eta_{min}, \eta_{max}]$, and set

$$Q_{k+1} = \eta_k Q_k + 1, C_{k+1} = (\eta_k Q_k C_k + f(k+1))/Q_{k+1} \tag{7}$$

Step5 Set $x_{k+1} = x_k + \alpha_k d_k$.

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

Note that the algorithm cannot cycle indefinitely in step 3.2 in fact, when $\alpha_k = 0$, $f(x_k + \alpha_k d_k) = f(x_k)$ there must exist a sufficient small α_k

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

because of $\nabla f(x_k) d_k \leq 0$ and $0 < \delta < 1$. what is more, $f(x_k) \leq C(x_k)$, So we have

$$f(x_k + \alpha_k d_k) \leq C_k + \delta \alpha_k \nabla f(x_k) d_k,$$

3 The Global Convergence of the Algorithm

In this section, we discuss the global convergence property of algorithm with the new nonmonotone line search. In order to achieve the convergence of the algorithm, 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 (2)(5), 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 (2)(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

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

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

holds end.

Lemma 3.3 If $\nabla f(x_k)d_k \leq 0$ for each k, then for the iterates generated by the algorithm (2.1), we have $f(x_k) \leq C_k \leq A_k$ for each k. Moreover, if $\nabla f(x_k)d_k < 0$ and $f(x_k)$ is bounded from below, then there exists α_k satisfying Armijo conditions of the line search update.
Proof: Defining $D_k : R \to R$ by

$$D_k(t) = \frac{tC_{k-1} + f_k}{t+1},$$

we have

$$D_k^{'}(t) = \frac{C_{k-1} - f_k}{(t+1)^2},$$

Since $\nabla f(x_k)d_k \leq 0$, it follows from (6) that $f_k \leq C_{k-1}$, which implies that $D_k^{'} \geq 0$ for all $t \geq 0$, Hence, D_k is nondecreasing, and $f_k = D_k(0) \leq D_k(k)$ for all $t \geq 0$. in particular, taking $t = \eta_{k-1}Q_{k-1}$ gives

$$f_k = D_k(0) \leq D_k(\eta_{k-1}Q_{k-1}) = C_k.$$

the upper bound $C_k \leq A_k$ is proved by induction. For $k = 0$ this holds by initialization $C_0 = f(x_0)$. Now assume that $C_j \leq A_j$ for all $0 \leq j < k$. by (4), the initialization $Q_0 = 1$, and the fact that $\eta_k \in [0,1]$, we have

$$Q_{j+1} = 1 + \sum_{i=0}^{j} \prod_{m=0}^{i} \eta_{j-m} \leq j + 2. \tag{8}$$

Since D_k is monotone nondecreasing, (8) imply that

$$C_k = D_k(\eta_{k-1}Q_{k-1}) = D_k(Q_k - 1) \leq D_k(k). \tag{9}$$

By the induction step,

$$D_k(k) = \frac{kC_{k-1} + f_k}{k+1} \leq \frac{kA_{k-1} + f_k}{k+1} = A_k, \tag{10}$$

Relation (9) and (10),we have the upper bound of C_k.

In fact, when $\alpha_k = 0$, $f(x_k + \alpha_k d_k) = f(x_k)$ there must exist a sufficient small α_k

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

because of $\nabla f(x_k)d_k \leq 0$ and $0 < \delta < 1$. what is more, $f(x_k) \leq C(x_k)$, So we have

$$f(x_k + \alpha_k d_k) \leq C_k + \delta\alpha_k \nabla f(x_k)d_k.$$

Lemma 3.4 As we are known, α_k is generated by step(3.2), so α_k satisfies (6), If assumption 3.1 holds, we have

$$\alpha_k \geq \frac{2\beta(1-\delta)|g_k^T d_k|}{L\|d_k\|^2}. \tag{11}$$

Proof: by algorithm 2.1 we have $\alpha_k \leq 1$, and $\alpha_k = \alpha_k \beta^{h_k}$, h_k is the smallest integer such that (6) holds. since we have

$$f(x_k + \frac{\alpha_k}{\beta} d_k) > C_k + \delta \frac{\alpha_k}{\beta} g_k^T d_k \geq f(x_k) + \delta \frac{\alpha_k}{\beta} g_k^T d_k \tag{12}$$

while ∇f is Lipschitz continuous,

$$\begin{aligned} f(x_k + \alpha d_k) - f(x_k) =& \alpha g_k^T d_k + \int_0^\alpha [\nabla f(x_k + t d_k) - \nabla f(x_k)] d_k dt \\ \leq& \alpha g_k^T d_k + \int_0^\alpha t L \|d_k\|^2 dt \\ =& \alpha g_k^T d_k + \frac{1}{2} L \alpha^2 \|d_k\|^2. \end{aligned}$$

Combining this with (12) we can proves (11) holds.
Theorem 3.1 The iterates x_k generated by the algorithm 2.1, then we have the property that

$$\lim_{k \to \infty} \nabla f(x_k) = 0. \tag{13}$$

Hence, every convergent subsequence of the iterates approaches a point x^*, where $\nabla f(x^*) = 0$.
Proof: we first show that

$$f_{k+1} \leq C_k - \varphi \|g_k\|^2, \tag{14}$$

where

$$\varphi = \frac{2\delta\beta(1-\delta)(1-\eta)^2}{L(1+\eta)^2},$$

by step 3.2 we known $\alpha \leq 1$ and by (11)

$$\alpha_k \geq \frac{2\beta(1-\delta)|g_k^T d_k|}{L\|d_k\|^2}.$$

and by (6) we have

$$f_{k+1} \leq C_k - \frac{2\delta\beta(1-\delta)}{L} (\frac{g_k^T d_k}{\|d_k\|})^2$$

Finally, with lemma3.1 lemma3.2

$$f_{k+1} \leq C_k - (\frac{2\delta\beta(1-\delta)(1-\eta)^2}{L(1+\eta)^2}) \|g_k\|^2$$

which implies (14) Combine the cost update (7) and upper bound (14)

$$\begin{aligned} C_{k+1} =& \frac{\eta_k Q_k C_k + f_{k+1}}{Q_{k+1}} \\ \leq& \frac{\eta_k Q_k C_k + C_k - \varphi \|g_k\|^2}{Q_{k+1}} \\ =& C_k - \frac{\varphi \|g_k\|^2}{Q_{k+1}} \end{aligned} \tag{15}$$

Since f is bounded from below and $f_k \leq C_k$ for all k, we can conclude that C_k is bounded from below. It follows from (15) that

$$\sum_{k=0}^{\infty} \frac{\|g_k\|^2}{Q_{k+1}} < \infty. \tag{16}$$

because of $\eta_{max} < 1$,then by (8) we have

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

Consequently, (16) implies

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

4 Numerical Results

To test the effect of the algorithm, we selected a few examples of numerical experiments, Although our best convergence result were obtained by dynamically varying η_k, using valuea closer to 1 when the iterates were far from the optimum, and using values closer to 0 when the iterates were near an optimum, in this numerical test we take some fixed value η_k, which seemed to work reasonably well for a broad class of problems.

Typical values for the others parameters are: $\delta = 0.75$, $\beta = 0.5$, we compute the value $f(\widehat{x})$ of the objective function at some fixed iterate step, we denote the value 3.100×10^{-3} by $3.100(-3)$, the algorithm has been tested on the following set of problems.

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

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

Problem 1	$\eta = 0$	$\eta = 0.1$	$\eta = 0.2$	$\eta = 0.3$
n=0	6.000	6.000	6.000	6.000
n=10	3.100(−3)	5.146(−5)	7.405(−4)	1.050(−2)
n=20	1.965(−8)	5.446(−10)	3.170(−8)	5.235(−8)
n=25	3.291(−11)	4.268(−12)	2.693(−10)	5.469(−9)

Problem 1	$\eta = 0.5$	$\eta = 0.6$	$\eta = 0.8$	$\eta = 0.9$
n=0	6.000	6.000	6.000	6.000
n=10	1.090(−2)	2.860(−2)	1.359(−1)	2.612(−1)
n=20	8.167(−5)	8.117(−4)	4.101(−2)	9.110(−2)
n=25	2.567(−5)	9.947(−5)	3.900(−3)	6.360(−2)

$$Problem2. f(x) = (x_1+10*x_2)^4+5*(x_3-x_4)^4+(x_2-2*x_3)^4+10*(x_1-10*x_4)^4$$

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

Problem 2	$\eta = 0$	$\eta = 0.1$	$\eta = 0.2$	$\eta = 0.3$
n=0	2.381(5)	2.381(5)	2.381(5)	2.381(5)
n=20	1.595(1)	7.980(−2)	2.200(−3)	3.859(−4)
n=30	2.860(−2)	6.590(−5)	1.354(−5)	2.856(−6)
n=40	1.146(−4)	1.411(−5)	7.357(−6)	8.602(−7)

Problem 2	$\eta = 0.5$	$\eta = 0.6$	$\eta = 0.8$	$\eta = 0.9$
n=0	2.381(5)	2.381(5)	2.381(5)	2.381(5)
n=20	9.995(−1)	1.515(−1)	1.408(1)	2.517(2)
n=30	1.113(−4)	1.505(−5)	3.045(2)	4.282(2)
n=40	2.026(−6)	7.756(−6)	1.540(−2)	2.382(2)

$$Problem3. f(x) = (x_1-1)^2+(x_1-x_2)^2+(x_3-1)^2+(x_4-1)^4+(x_5-1)^6$$

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

Problem 3	$\eta = 0$	$\eta = 0.1$	$\eta = 0.2$	$\eta = 0.3$
n=0	4.000	4.000	4.000	4.000
n=10	3.360(−2)	1.100(−3)	1.100(−3)	3.600(−3)
n=20	5.475(−4)	2.794(−4)	2.114(−4)	7.032(−6)
n=30	5.475(−4)	1.268(−4)	6.998(−5)	5.730(−6)

Problem 3	$\eta = 0.5$	$\eta = 0.6$	$\eta = 0.8$	$\eta = 0.9$
n=0	4.000	4.000	4.000	4.000
n=10	5.700(−3)	7.020(−2)	1.232(−1)	2.593(−1)
n=20	1.739(−4)	1.200(−3)	1.880(−2)	8.340(−2)
n=30	1.696(−5)	2.452(−5)	2.800(−3)	2.640(−2)

$$Problem 4. f(x) = (1-x_1)^2 + (1-x_{10})^2 + \sum_{i=1}^{9}(x_i^2 - x_{i+1})^2$$

$$x_0 = (3, \cdots, 3)^T, x^* = (1, \cdots, 1)^T, f^* = 0.$$

Problem 4	$\eta = 0$	$\eta = 0.1$	$\eta = 0.2$	$\eta = 0.3$
n=0	3.320(2)	3.320(2)	3.320(2)	3.320(2)
n=20	6.189(−4)	2.020(−4)	1.200(−3)	5.278(−1)
n=30	9.862(−7)	9.089(−7)	1.551(−5)	5.109(−1)
n=50	1.738(−11)	5.116(−11)	1.082(−9)	5.060(−1)

Problem 4	$\eta = 0.5$	$\eta = 0.6$	$\eta = 0.8$	$\eta = 0.9$
n=0	3.320(2)	3.320(2)	3.320(2)	3.320(2)
n=20	5.300(−1)	5.738(−1)	1.341	4.767
n=30	5.136(−1)	5.169(−1)	7.131(−1)	2.502
n=50	5.066(−1)	5.080(−1)	5.345(−1)	8.510(−1)

When $\eta = 0$ the algorithm take monotone Armijo line search, which is the result of [12], from the tables we can conclude: Although not all nonmonotone line search is performing better then monotone line search, there exist the choice of η ensures the best convergence of our algorithm, especially the choice of $\eta = 0.1$, The numerical performance is very dependent on the choice of η. furthermore, there are some advantages In theory, the best convergence result were obtained by dynamically varying η_k, using valuea closer to 1 when the iterates were far from the optimum, and using values closer to 0 when the iterates were near an optimum.

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] 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.

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

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

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

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

[12] Jingyong. Tang, and Li. Dong, A new memory gradient method and its convergence, 2009, Vol 27.

[13] Hongchao Zhang and William W. Hager. A nonmonotone line search technique and its application to unconstrained optimization, SIAM J. Optim. 14, 2004, 1043-1056.

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

[15] 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.

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

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

[18] L. Grippo, F. Lampariello, and S. Lucidi, A class of nonmonotone stabilization methods in unconstrained optimization, numer. Math., 59, 1991, 779-805.

Received: August, 2015