SUT Journal of Mathematics
Vol. 42, No. 2 (2006), 191–206

A memory gradient method without line search for unconstrained optimization

Yasushi Narushima

(Received July 24, 2006)

Abstract. Memory gradient methods are used for unconstrained optimization, especially large scale problems. The first idea of memory gradient methods was proposed by Miele and Cantrell (1969) and subsequently extended by Cragg and Levy (1969). Recently Narushima and Yabe (2006) proposed a new memory gradient method which generates a descent search direction for the objective function at every iteration and converges globally to the solution if the Wolfe conditions are satisfied within the line search strategy. On the other hand, Sun and Zhang (2001) proposed a particular choice of step size, and they applied it to the conjugate gradient method. In this paper, we apply the choice of the step size proposed by Sun and Zhang to the memory gradient method proposed by Narushima and Yabe and establish its global convergence.

AMS 2000 *Mathematics Subject Classification.* 90C06, 90C30, 65K05.

Key words and phrases. Nonlinear programming, optimization, memory gradient method, global convergence, large scale problems.

§1. Introduction

We consider the following unconstrained optimization problem

$$\text{minimize} \qquad f(x), \tag{1.1}$$

where $f:\ \mathbf{R}^n \to \mathbf{R}$ is sufficiently smooth and its gradient $g \equiv \nabla f$ is available. We denote $g(x_k)$ by g_k and the Euclidean norm by $\|\cdot\|$. Usually we use the iterative method for solving the problem (1.1) and its form is given by

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

where $x_k \in \mathbf{R}^n$ is the k-th approximation to the solution, $\alpha_k \in \mathbf{R}$ is a step size and $d_k \in \mathbf{R}^n$ is a search direction.

There exist many kinds of iterative methods. In general, the Newton method and quasi-Newton methods are very effective to solve problem (1.1). These methods, however, must keep matrices of size $n \times n$. Thus these methods cannot always be applied to large scale problems. Although the steepest descent method does not need any matrices, it has slow rate of convergence. Accordingly, acceleration of the steepest descent method (which does not need any matrices) has recently attracted attention. For instance, the conjugate gradient method is one of the most famous methods. The search direction of this method is usually defined by

$$d_k = -g_k + \beta_k d_{k-1}, \tag{1.3}$$

where $\beta_k \in \mathbf{R}$. The parameter β_k is chosen so that the method (1.2)–(1.3) reduces to the linear conjugate gradient method if $f(x)$ is a strictly convex quadratic function and if α_k is the exact one-dimensional minimizer. Well-known formulas for β_k are the Fletcher-Reeves (FR), Polak-Ribiére-Polyak (PRP), Hestenes-Stiefel (HS) and Dai-Yuan (DY) formulas, and they are given by

$$\begin{aligned} \beta_k^{FR} &= \|g_k\|^2/\|g_{k-1}\|^2, \\ \beta_k^{PRP} &= g_k^T y_{k-1}/\|g_{k-1}\|^2, \\ \beta_k^{HS} &= g_k^T y_{k-1}/d_{k-1}^T y_{k-1}, \\ \beta_k^{DY} &= \|g_k\|^2/d_{k-1}^T y_{k-1}, \end{aligned}$$

where $y_{k-1} = g_k - g_{k-1}$. The global convergence properties of the conjugate gradient methods have been studied by many researchers (see [3, 9] for example).

The memory gradient method also aims to accelerate the steepest descent method and it was first proposed by Miele and Cantrell [7] and was subsequently extended by Cragg and Levy [2]. The search direction of this method is defined by

$$d_k = -\gamma_k g_k + \sum_{i=1}^{m} \xi_{ki} d_{k-i},$$

where m is the number of past iterations remembered, $\xi_{ki} \in \mathbf{R}$ $(i = 1, \ldots, m)$ and $\gamma_k \in \mathbf{R}$ are parameters. More recently, a different type of memory gradient methods were proposed by Narushima and Yabe [11]. These methods always satisfy the sufficient descent condition and converge globally if the Wolfe conditions are satisfied within the line search strategy. Moreover Narushima [10]

combined it with nonmonotone line search strategy and established the global convergence.

It is important to study how we choose a step size in iterative methods. Usually we choose a step size which satisfies the Wolfe conditions

$$f(x_k) - f(x_k + \alpha_k d_k) \geq -\sigma_1 \alpha_k g_k^T d_k, \tag{1.4}$$

$$g(x_k + \alpha_k d_k)^T d_k \geq \sigma_2 g_k^T d_k, \tag{1.5}$$

or the Armijo condition (1.4) only, where $0 < \sigma_1 < \sigma_2 < 1$. In those line search techniques, it is necessary to compute the function and the gradient value several times at each iteration. For very large scale problems, these computations can be too expensive.

Sun and Zhang [12] proposed a particular choice of step size, which means no line search. They gave the following step size:

$$\alpha_k = -\delta \frac{g_k^T d_k}{d_k^T Q_k d_k},$$

where δ is some positive constant and $\{Q_k\}$ is a sequence of symmetric positive definite matrices. In addition, they established global convergence of some conjugate gradient methods without line search. There are some applications which use the above step size [1, 5].

In the present paper, we will consider a memory gradient method, which was proposed by Narushima and Yabe [11], without line search and prove its global convergence.

This paper is organized as follows. In Section 2, we analyze general iterative methods without line search and consider a sufficient condition for the global convergence. In Section 3, we apply the method in Section 2 to the memory gradient method proposed by Narushima and Yabe [11], and prove its global convergence. In Section 4, we propose one choice of $\{Q_k\}$. In Section 5, some numerical results are reported and conclusions are made in Section 6.

§2. General iterative method without line search

In this section, we discuss iterative methods with no line search which is given by Sun and Zhang [12].

First we introduce the choice of the step size proposed in [12]. Let $\{Q_k\}$ be a sequence of symmetric and uniformly positive definite matrices, namely, there exist positive constants $\nu_{\min}$ and $\nu_{\max}$ such that

$$\nu_{\min}\|v\|^2 \leq v^T Q_k v \leq \nu_{\max}\|v\|^2 \tag{2.1}$$

for all $v \in \mathbf{R}^n$ and all k. We use the following step size proposed in [12]

$$\alpha_k = -\delta \frac{g_k^T d_k}{d_k^T Q_k d_k}, \tag{2.2}$$

where δ is a positive constant. In this paper, we call this step size *Sun-Zhang's step size*. We emphasize that d_k in (2.2) is allowed to be any nonzero search direction with $g_k^T d_k \neq 0$. Usually we expect that d_k is descent, but this formula allows us that d_k is even ascent. Specifically, whether d_k is a descent direction or not, $\alpha_k d_k$ becomes a descent direction, i.e., $g_k^T(\alpha_k d_k) < 0$ as long as $g_k^T d_k \neq 0$. If $g_k^T d_k = 0$, then we can use $d_k = -g_k$ and $\alpha_k = \delta g_k^T g_k / g_k^T Q_k g_k$, for example.

Now we introduce the algorithm of general iterative methods without line search.

Algorithm 2.1. (**General iterative method without line search**)

Step 0. *Given* $x_0 \in \mathbf{R}^n$. *Set* $k := 0$.

Step 1. *Compute a search direction* d_k.

Step 2. *Compute a step size* α_k *by* (2.2).

Step 3. *Let* $x_{k+1} = x_k + \alpha_k d_k$. *If a stopping criterion is satisfied, then stop.*

Step 4. *Set* $k := k + 1$ *and go to Step* 1.

Next, in order to establish the subsequent theorems, we make the following assumptions.

Assumption 2.2.

(A1) *The objective function* f *is bounded below on* $\mathbf{R}^n$ *and is continuously differentiable in a convex neighborhood* $\mathcal{N}$ *of the level set* $\mathcal{L} = \{x \in \mathbf{R}^n : f(x) \leq f(x_0)\}$ *at the initial point* x_0.

(A2) *The convex neighborhood* $\mathcal{N}$ *includes the sequence* $\{x_k\}$ *generated by Algorithm* 2.1, *namely,* $\{x_k\} \subset \mathcal{N}$.

(A3) *The gradient* g *is Lipschitz continuous in* $\mathcal{N}$, *i.e., there exists a positive constant* L *such that*

$$\|g(x) - g(y)\| \leq L\|x - y\|$$

for all $x, y \in \mathcal{N}$.

It should be noted that the assumption that the objective function is bounded below is weaker than the usual assumption that the level set is bounded.

Now we consider a sufficient condition which establishes the global convergence. In the rest of this section, we assume $g_k \neq 0$ for all k, otherwise a stationary point has been found. The following lemma is proved by Sun and Zhang [Lemma 4, 12].

Lemma 2.3. *Suppose that Assumption* 2.2 *is satisfied. Let* $\{x_k\}$ *be a sequence generated by Algorithm* 2.1 *with* $\delta \in (0,\ \nu_{\min}/L)$. *Then the sequence* $\{f(x_k)\}$ *is non-increasing and the following holds:*

$$\sum_{k=0}^{\infty} \frac{(g_k^T d_k)^2}{\|d_k\|^2} < \infty.$$

Note that Sun and Zhang [12] assume the boundedness of the level set, but it is unnecessary for this lemma.

We are interested in the condition under which we establish the global convergence property. To this end, we consider the cosine measure

$$\cos\theta_k = -\frac{g_k^T(\alpha_k d_k)}{\|g_k\|\|\alpha_k d_k\|} = \frac{|g_k^T d_k|}{\|g_k\|\|d_k\|}.$$

This measure is the cosine of the angle between $\alpha_k d_k$ and the steepest descent direction $-g_k$.

The next theorem means that the sequence $\{x_k\}$ generated by Algorithm 2.1 converges if there is a subsequence $\{x_{k'}\}$ of $\{x_k\}$ such that $\cos\theta_{k'}$ is bounded away from zero for k' sufficiently large.

Theorem 2.4. *Suppose that all assumptions of Lemma* 2.3 *hold and there exist a positive constant* c_1 *and a subsequence* $\{x_{k'}\}$ *of* $\{x_k\}$ *such that* $\cos\theta_{k'} \geq c_1$ *for all* k' *sufficiently large. Then the sequence* $\{x_k\}$ *converges in the sense that*

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

Proof. If the theorem is not true, there exists a constant $c_2 > 0$ such that

$$\|g_k\| \geq c_2 \tag{2.3}$$

for all k. Then from (2.3) and the assumption $\cos\theta_{k'} \geq c_1$, we have

$$\frac{|g_{k'}^T d_{k'}|}{\|d_{k'}\|} = \frac{\|g_{k'}\|\|d_{k'}\|\cos\theta_{k'}}{\|d_{k'}\|} \geq c_1 c_2$$

for all k' sufficiently large. Therefore, we obtain

$$\sum_{k'}^{\infty} \frac{(g_{k'}^T d_{k'})^2}{\|d_{k'}\|^2} = \infty,$$

which contradicts Lemma 2.3. Therefore the proof is complete. □

We next consider the sufficient descent condition, namely, for some constant $c_3 > 0$,

$$g_k^T d_k \leq -c_3 \|g_k\|^2 \tag{2.4}$$

for all k. The sufficient descent condition is a stronger condition than the descent condition $g_k^T d_k < 0$. We sometimes assume it to analyze convergence properties of iterative methods. The following proposition implies that the sufficient descent condition holds if $\cos\theta_k$ is bounded away from zero.

Proposition 2.5. *Suppose that Assumption* 2.2 *holds. Let the sequence* $\{x_k\}$ *be generated by Algorithm* 2.1. *If there exists a positive constant* $\hat{c}_1$ *such that* $\cos\theta_k \geq \hat{c}_1$ *for all* k, *then* $\alpha_k d_k$ *satisfies the sufficient descent condition, namely, there exists some positive constant* $\hat{c}_3$ *such that*

$$g_k^T(\alpha_k d_k) \leq -\hat{c}_3 \|g_k\|^2$$

for all k.

Proof. From (2.2), (2.1) and $\cos\theta_k \geq \hat{c}_1$, we have

$$\begin{aligned}
\alpha_k g_k^T d_k &= -\delta \frac{(g_k^T d_k)^2}{d_k^T Q_k d_k} \\
&\leq -\frac{\delta}{\nu_{\max}} \frac{(g_k^T d_k)^2}{\|d_k\|^2} \\
&= -\frac{\delta}{\nu_{\max}} \frac{\|g_k\|^2 \|d_k\|^2 \cos^2\theta_k}{\|d_k\|^2} \\
&\leq -\frac{\delta \hat{c}_1^2}{\nu_{\max}} \|g_k\|^2.
\end{aligned}$$

This implies that the sufficient descent condition holds with $\hat{c}_3 = \delta \hat{c}_1^2 / \nu_{\max}$. □

§3. The memory gradient method without line search

In this section, we combine Sun-Zhang's step size (2.2) with the memory gradient method proposed by Narushima and Yabe [11]. We define a search direction by the form

$$d_k = -\gamma_k g_k + \frac{1}{m} \sum_{i=1}^{m} \beta_{ki} d_{k-i}, \qquad (k \geq 1) \tag{3.1}$$

where $\beta_{ki} \in \mathbf{R}$ $(i = 1, \ldots, m)$, $\gamma_k \in [\underline{\gamma},\ \bar{\gamma}]$ are parameters, and $\underline{\gamma}$ and $\bar{\gamma}$ are given positive constants. Note that for the case $k < m$, equation (3.1) is

interpreted as $d_k = -\gamma_k g_k + \dfrac{1}{k}\sum_{i=1}^{k} \beta_{ki} d_{k-i}$. The search direction at the first iteration is the steepest descent direction with a sizing parameter $\gamma_0 > 0$, namely, $d_0 = -\gamma_0 g_0$. We define β_{ki} as follows:

$$\beta_{ki} = \|g_k\|^2 \psi_{ki}^{\dagger}, \tag{3.2}$$

where $a^{\dagger}$ is defined by

$$a^{\dagger} = \begin{cases} 0 & \text{if } a = 0, \\ \dfrac{1}{a} & \text{otherwise,} \end{cases}$$

and ψ_{ki} $(i = 1, \ldots, m)$ are parameters which satisfy the condition

$$\begin{cases} g_k^T d_{k-1} + \|g_k\|\|d_{k-1}\| & < \gamma_k \psi_{k1} \qquad (i = 1), \\ g_k^T d_{k-i} + \|g_k\|\|d_{k-i}\| & \le \gamma_k \psi_{ki} \qquad (i = 2, \ldots, m). \end{cases} \tag{3.3}$$

Note that $\beta_{k1} > 0$ and $\beta_{ki} \ge 0$ $(i = 2, \ldots, m)$ hold by the fact that $\psi_{k1} > 0$ and $\psi_{ki} \ge 0$ $(i = 2, \ldots, m)$. It is known that the memory gradient method with (3.1)–(3.3) always satisfies the descent condition. The next lemma was given by Narushima and Yabe [Theorem 2.1, 11].

Lemma 3.1. *Let d_k be defined by the memory gradient method* (3.1)–(3.3). *Then d_k satisfies the descent condition $g_k^T d_k < 0$ for all k.*

By using Theorem 2.4 and Lemma 3.1, we obtain the following theorem.

Theorem 3.2. *Suppose all assumptions of Lemmas* 2.3 *and* 3.1 *hold. Then $\{x_k\}$ achieves a solution in a finite number of iterations or converges in the sense that*

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

Proof. If the algorithm does not terminate after finite many iterations, we have that

$$\|g_k\| > 0 \quad \text{for all } k.$$

From (3.1), we have

$$\|d_k\|^2 = \left\| \frac{1}{m} \sum_{i=1}^{m} \beta_{ki} d_{k-i} \right\|^2 - 2\gamma_k g_k^T d_k - \gamma_k^2 \|g_k\|^2.$$

Dividing both sides by $(g_k^T d_k)^2$, we obtain that

$$\begin{aligned}
\frac{\|d_k\|^2}{(g_k^T d_k)^2} &= \frac{\|\frac{1}{m}\sum_{i=1}^m \beta_{ki} d_{k-i}\|^2}{(g_k^T d_k)^2} - 2\gamma_k \frac{g_k^T d_k}{(g_k^T d_k)^2} - \gamma_k^2 \frac{\|g_k\|^2}{(g_k^T d_k)^2} \\
&= \frac{\|\frac{1}{m}\sum_{i=1}^m \beta_{ki} d_{k-i}\|^2}{(g_k^T d_k)^2} - \gamma_k \frac{2}{g_k^T d_k} - \gamma_k^2 \frac{\|g_k\|^2}{(g_k^T d_k)^2} \\
&= \frac{\|\frac{1}{m}\sum_{i=1}^m \beta_{ki} d_{k-i}\|^2}{(g_k^T d_k)^2} - \left(\frac{1}{\|g_k\|} + \gamma_k \frac{\|g_k\|}{g_k^T d_k}\right)^2 + \frac{1}{\|g_k\|^2} \\
&\leq \frac{\|\frac{1}{m}\sum_{i=1}^m \beta_{ki} d_{k-i}\|^2}{(g_k^T d_k)^2} + \frac{1}{\|g_k\|^2} \\
&\leq \left(\frac{\frac{1}{m}\sum_{i=1}^m \beta_{ki}\|d_{k-i}\|}{|g_k^T d_k|}\right)^2 + \frac{1}{\|g_k\|^2}. \qquad (3.4)
\end{aligned}$$

On the other hand, we obtain from Lemma 3.1, (3.1), (3.2), (3.3) and the fact that $\psi_{ki}^\dagger \psi_{ki} \leq 1$

$$\begin{aligned}
|g_k^T d_k| &= -g_k^T d_k \\
&= \gamma_k \|g_k\|^2 - \frac{1}{m}\sum_{i=1}^m \beta_{ki} g_k^T d_{k-i} \\
&= \frac{1}{m}\sum_{i=1}^m (\gamma_k \|g_k\|^2 - \beta_{ki} g_k^T d_{k-i}) \\
&\geq \frac{1}{m}\sum_{i=1}^m (\gamma_k \|g_k\|^2 \psi_{ki}^\dagger \psi_{ki} - \beta_{ki} g_k^T d_{k-i}) \\
&\geq \frac{1}{m}\sum_{i=1}^m (\gamma_k \psi_{ki} - g_k^T d_{k-i})\beta_{ki} \\
&\geq \frac{1}{m}\|g_k\| \sum_{i=1}^m \beta_{ki}\|d_{k-i}\|. \qquad (3.5)
\end{aligned}$$

The last inequality follows from the fact that $\gamma_k \psi_{ki} - g_k^T d_{k-i} \geq \|g_k\|\|d_{k-i}\|$ yields

$$\sum_{i=1}^m \beta_{ki}(\gamma_k \psi_{ki} - g_k^T d_{k-i}) \geq \|g_k\| \sum_{i=1}^m \beta_{ki}\|d_{k-i}\|.$$

Therefore we have from (3.5)

$$\frac{\frac{1}{m}\sum_{i=1}^m \beta_{ki}\|d_{k-i}\|}{|g_k^T d_k|} \leq \frac{1}{\|g_k\|}. \qquad (3.6)$$

Finally we obtain from (3.4) and (3.6)

$$\frac{(g_k^T d_k)^2}{\|d_k\|^2} \geq \frac{\|g_k\|^2}{2},$$

which implies that $\cos\theta_k \geq \frac{1}{\sqrt{2}}$. Therefore from Theorem 2.4, the proof is complete. □

§4. Choice of matrix Q_k

In this section, we give a concrete choice of Q_k. Sun-Zhang's step size (2.2) can be interpreted as a minimizer of the quadratic model $F(\alpha)$ of $f(x_k + \alpha d_k)$ in α

$$F(\alpha) = f(x_k) + \alpha g_k^T d_k + \frac{\alpha^2}{2} d_k^T B_k d_k \approx f(x_k + \alpha d_k),$$

where B_k is $\nabla^2 f(x_k)$ or its approximation. From $F'(\alpha) = 0$, we have (2.2) with $\delta = 1$ and $Q_k = B_k$. Therefore it is appropriate that Q_k is an approximation matrix to the Hessian matrix $\nabla^2 f(x_k)$. To generate the symmetric positive definite approximation matrix, the BFGS or the DFP updating formula is usually used. However the matrix updated by the BFGS formula is not necessarily positive definite when the inequality $s_{k-1}^T y_{k-1} > 0$ is not satisfied, where $s_{k-1} = x_k - x_{k-1}$ and $y_{k-1} = g_k - g_{k-1}$. In order to overcome this weakness, Li and Fukushima [6] proposed the modified BFGS update

$$B_k = B_{k-1} - \frac{B_{k-1} s_{k-1} s_{k-1}^T B_{k-1}}{s_{k-1}^T B_{k-1} s_{k-1}} + \frac{z_{k-1} z_{k-1}^T}{s_{k-1}^T z_{k-1}}, \tag{4.1}$$

where

$$z_{k-1} = y_{k-1} + \lambda_{k-1} s_{k-1} \tag{4.2}$$

and λ_{k-1} is a nonnegative parameter such that $s_{k-1}^T z_{k-1} > 0$. If B_{k-1} is positive definite, then the modified BFGS update always generates the positive definite approximation matrix. However we must store the matrix if we use (4.1) as Q_k. Thus we recommend the formula

$$Q_k = \eta_k I - \eta_k \frac{s_{k-1} s_{k-1}^T}{s_{k-1}^T s_{k-1}} + \frac{z_{k-1} z_{k-1}^T}{s_{k-1}^T z_{k-1}}, \tag{4.3}$$

where η_k is a positive sizing parameter and I denotes the unit matrix. The above formula is the modified BFGS update (4.1) with $B_{k-1} = \eta_k I$. When we use (4.3) as Q_k, we can compute $d_k^T Q_k d_k$ without matrix-vector product and do not need keeping any matrices.

§5. Numerical results

In previous sections, we establish the global convergence of the memory gradient method with Sun-Zhang's step size. In this section, we give some numerical results to investigate the practical performance of the proposed method. For this purpose, we first study the behavior of the sequence $\{f(x_k)\}$ and next discuss the results of our method for general test functions.

In our experiment, we first chose γ_k and next determined ψ_{ki} $(i = 1, \ldots, m)$ that satisfy condition (3.3). We chose $\gamma_0 = 1$ and

$$\gamma_k = \frac{z_{k-1}^T s_{k-1}}{z_{k-1}^T z_{k-1}}$$

for $k \geq 1$, where z_{k-1} is defined by (4.2). Though this choice of the sizing parameter is different from the sizing parameter used in [10, 11], it is natural to choose such a parameter, because z_{k-1} is used instead of y_{k-1} in updating Q_k. Moreover we used $\eta_0 = 1$ and

$$\eta_k = \frac{z_{k-1}^T s_{k-1}}{s_{k-1}^T s_{k-1}}$$

for $k \geq 1$. For given γ_k, we used ψ_{ki} $(i = 1, \ldots, m)$ defined by

$$\psi_{ki} = \frac{||g_k||||d_{k-i}|| + g_k^T d_{k-i} + n}{\gamma_k}.$$

In order to establish $s_{k-1}^T z_{k-1} > 0$, we set

$$\lambda_{k-1} = \begin{cases} 0 & s_{k-1}^T y_{k-1} > 0, \\ 2^i & \text{otherwise}, \end{cases}$$

where i is the smallest integer such that $s_{k-1}^T z_{k-1} > 0$ holds. The stopping condition was

$$\|g_k\| \leq 10^{-5}.$$

To investigate the behavior of the sequence $\{f(x_k)\}$, we performed our method for two-dimensional functions. For two-dimensional functions, we chose $(2, 3)^T$ as a starting point and set $m = 3$ and $\delta = 1$ or $\delta = 0.099$. We set $\alpha_0 = \delta$ and α_k was computed by (2.2) with (4.3). Figures 1–6 give the values of $\log_{10}(f(x) - f(x^*))$, where x^* is the solution of each problem. The first test function is the following strictly convex quadratic function

$$f(x, y) = \begin{bmatrix} x \\ y \end{bmatrix}^T A \begin{bmatrix} x \\ y \end{bmatrix},$$

where $A = \begin{bmatrix} 10 & 0 \\ 0 & 1 \end{bmatrix}$. Since the matrix A has eigen-values $\nu_{\max} = 10$ and $\nu_{\min} = 1$, we have $\nu_{\min}/\nu_{\max} = 0.1$. We note that $0.099 \in (0,\ \nu_{\min}/L)$ and $1 \notin (0,\ \nu_{\min}/L)$. Our method with $\delta = 1$ converges faster than that with $\delta = 0.099$ does. From Figure 1, we see that the monotonicity of $\{f(x_k)\}$ can not be found when $\delta = 1$. From Figure 2, we observe the monotonicity of $\{f(x_k)\}$ when $\delta = 0.099$. Next, we also investigate the behavior of the sequence $\{x_k\}$ when the objective function is the following non-quadratic function

$$f(x, y) = \cosh(x) + 2\cosh(y) + (xy)^2.$$

As well as the case of the quadratic function, our method with $\delta = 1$ converges faster than that with $\delta = 0.099$ does. From Figure 3, we see that $\{f(x_k)\}$ decreases monotonically except for $k = 0$, which is caused by $\alpha_0 = \delta = 1$. For the case $\delta = 0.099$, we also find the monotonicity of $\{f(x_k)\}$ from Figure 4. Moreover we examined the behavior for non-convex function

$$f(x) = \sum_{i=1,2} \left\{ i \left(\frac{1}{1 + e^{-x_i}} + \frac{1}{1 + e^{x_i}} \right) + x_i^2 \right\} + \prod_{i=1,2} x_i^2.$$

From Figure 5, we see that the monotonicity of $\{f(x_k)\}$ can be found except for $k = 0$ when $\delta = 1$. From Figure 6, we also see that the monotonicity of $\{f(x_k)\}$ can be found when $\delta = 0.099$. In the above three cases, we see that our method with $\delta = 1$ outperformed our method with $\delta = 0.099$. The parameter δ should be chosen not too much small if we can. However when the objective function is a general non-convex function, we cannot estimate $\nu_{\min}/L$ and cannot choose δ such that $\delta \in (0,\ \nu_{\min}/L)$. In this case, the proposed method might not converge.

In order to investigate robustness of our method, we performed our method for general test functions. In this experiment, the following three choices of α_k are used (called M1, M2, and M3, respectively):

M1. α_k chosen by (2.2) with (4.3).

M2. α_k chosen by (2.2) with the modified BFGS update (4.1).

M3. α_k chosen by the bisection line search method with the Armijo condition (1.4).

We set $\sigma_1 = 0.0001$ in the Armijo condition and $\delta = 1$ in Sun-Zhang's step size and set the initial matrix $Q_0 = I$ in M1 and M2. Although we examined our method with $Q_k = I$, it did not converge for almost all problems. So we do not present the results. In addition, we could not perform M2 for large scale problems, because the approximation matrix B_k is too big. We examined our method with $m = 1,\ 3,\ 5,\ 7,\ 9$.

In Table 1, the first column, the second column and the third column denote the problem number used in this paper, the problem name and the dimension of the problem, respectively. Problems P1 and P2 are defined by

Table 1: Test problems

P	Name	Dimension n
1	Quadratic function with "bcsstk02"	66
2	Quadratic function with "bcsstm02"	66
3	Extended Rosenbrock function	100 or 10000
4	Extended Powell Singular function	100 or 10000
5	Trigonometric function	100 or 10000
6	Broyden tridiagonal function	100 or 10000
7	Oren function	100
8	Cube function	2
9	Wood function	4
10	Beale function	2
11	Helical valley function	3
12	Jennrich and Sampson function	2

$$f(x) = x^T Ax + b^T x,$$

where $A \in \mathbf{R}^{n\times n}$ is a matrix and $b \in \mathbf{R}^n$ is a vector. We set the matrices A which are described in "Matrix Market" [13] ("bcsstk02" and "bcsstm02" are matrix name), b is the all one vector and starting point x_0 is the zero vector. Problems P1–P6 and P9–P12 are described by Moré et al. [8] and problems P7 and P8 are described in Grippo et al. [4]. Tables 2–4 give the numerical results of the form: (the number of iterations)/(the number of function value evaluations). We write "*Failed* " when the number of iterations exceeds 1000 and we write "*Failed** " when a numerical overflow occurs.

From Table 2, we see that there exist non-convergence cases (P3, P4, P8 and P9 for example). From Tables 2 and 3, we find that M1 is comparable with M2 in many problems but M1 is more robust than M2. Finally, comparing M1 with M3, we see that M3 outperformed M1 for many problems. However M1 outperformed M3 for some problems (see P5 and P6 in Tables 2 and 4, for instance).

Figure 1: The function value
($\delta = 1,\ m = 3$)

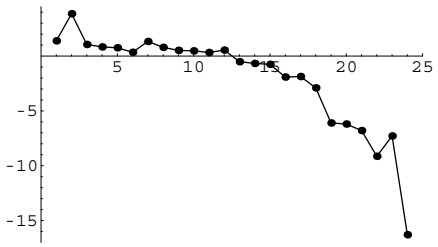

Figure 2: The function value
($\delta = 0.099,\ m = 3$)

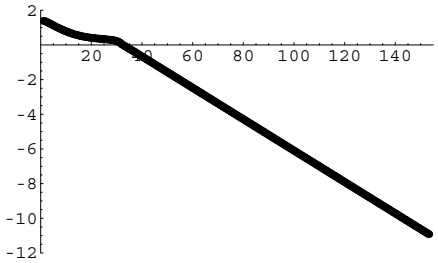

Figure 3: The function value
($\delta = 1,\ m = 3$)

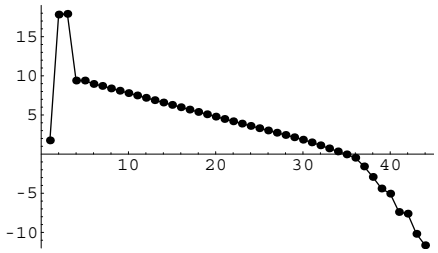

Figure 4: The function value
($\delta = 0.099,\ m = 3$)

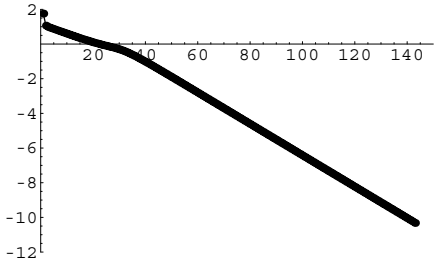

Figure 5: The function value
($\delta = 1,\ m = 3$)

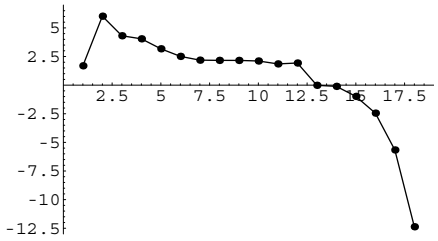

Figure 6: The function value
($\delta = 0.099,\ m = 3$)

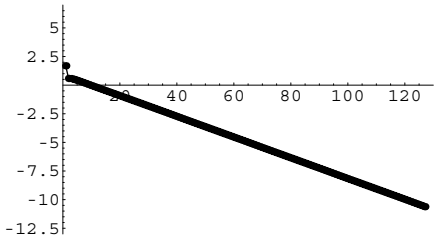

Table 2: Results of M1

P	n	$m = 1$	$m = 3$	$m = 5$	$m = 7$	$m = 9$
1	66	68/69	84/85	106/107	78/79	80/81
2	66	25/26	34/35	32/33	27/28	36/37
3	100	*Failed*	*Failed*	286/287	*Failed*	*Failed*
	10000	*Failed*	*Failed*	*Failed*	*Failed*	*Failed*
4	100	*Failed*	781/782	705/706	507/508	906/907
	10000	871/872	560/561	*Failed*	899/900	*Failed*
5	100	62/63	80/81	77/78	73/74	81/82
	10000	61/62	61/62	61/62	61/62	61/62
6	100	49/50	56/57	52/53	60/61	75/76
	10000	89/90	97/98	78/79	67/68	88/89
7	100	208/209	194/195	181/182	192/193	201/202
8	2	*Failed**	*Failed**	*Failed**	*Failed*	*Failed**
9	4	*Failed*	*Failed*	*Failed*	*Failed*	*Failed*
10	2	12/13	12/13	12/13	12/13	12/13
11	3	41/42	17/18	18/19	30/31	19/20
12	2	*Failed**	179/180	136/137	255/256	130/131

Table 3: Results of M2

P	n	$m = 1$	$m = 3$	$m = 5$	$m = 7$	$m = 9$
1	66	219/220	190/191	196/197	198/199	200/201
2	66	41/42	47/48	47/48	37/38	41/42
3	100	*Failed*	*Failed*	*Failed*	*Failed*	*Failed*
4	100	*Failed*	*Failed*	*Failed*	*Failed*	*Failed*
5	100	130/131	69/70	60/61	79/80	62/63
6	100	173/174	85/86	*Failed*	114/115	*Failed*
7	100	*Failed**	252/253	197/198	254/255	393/394
8	2	*Failed**	*Failed**	*Failed**	*Failed**	*Failed**
9	4	*Failed*	*Failed*	*Failed*	*Failed*	*Failed*
10	2	12/13	12/13	12/13	12/13	12/13
11	3	57/58	21/22	28/29	32/33	24/25
12	2	*Failed**	3/4	591/592	416/417	*Failed*

Table 4: Results of M3

P	n	$m=1$	$m=3$	$m=5$	$m=7$	$m=9$
1	66	36/48	38/50	42/54	46/58	35/47
2	66	23/24	26/27	28/29	26/27	25/26
3	100	89/145	84/125	80/129	90/147	58/100
	10000	94/157	85/137	80/129	85/149	58/100
4	100	227/357	181/281	175/285	166/287	221/361
	10000	244/403	222/358	230/399	244/415	213/359
5	100	85/92	88/92	82/88	80/87	80/87
	10000	69/72	68/69	68/69	68/70	68/70
6	100	97/1685	100/1648	103/1672	103/1576	110/1886
	10000	106/1925	111/2096	88/1463	97/1759	114/2116
7	100	235/331	178/238	173/238	171/236	157/214
8	2	44/75	39/67	46/75	54/108	56/109
9	4	202/280	95/137	122/173	106/159	151/216
10	2	8/13	9/14	9/14	8/13	8/13
11	3	18/38	16/28	17/29	25/88	24/104
12	2	19/35	22/41	18/39	25/49	20/43

§6. Conclusion

In this paper, we have combined the memory gradient method in [11] with Sun-Zhang's step size in [12] and have proved its global convergence property under the appropriate assumptions. Finally some numerical experiments have been shown. Our further interests are to study the convergence rate of the proposed method and to investigate new appropriate choices of parameters ψ_{ki} and δ.

Acknowledgements

The author would like to thank the referee for valuable comments. The author is grateful to Professor Hiroshi Yabe of Tokyo University of Science for his valuable advice and encouragement. The author would like to thank Dr. Hideho Ogasawara of Tokyo University of Science for valuable comments.

References

[1] X. Chen and J. Sun, Global convergence of a two-parameter family of conjugate gradient methods without line search, *Journal of Computational and Applied*

Mathematics, **146** (2002), 37–45.

[2] E. E. Cragg and A. V. Levy, Study on a supermemory gradient method for the minimization of functions, *Journal of Optimization Theory and Applications*, **4** (1969), 191–205.

[3] D. Y. Dai and Y. Yuan, A nonlinear conjugate gradient method with a strong global convergence property, *SIAM Journal on Optimization*, **10** (1999), 177–182.

[4] L. Grippo, F. Lampariello, and S. Lucidi, A truncated Newton method with nonmonotone line search for unconstrained optimization, *Journal of Optimization Theory and Applications*, **60** (1989), 401–419.

[5] X. Li and X. Chen, Global convergence of shortest-residual family of conjugate gradient methods without line search, *Asia-Pacific Journal of Operational Research*, **22** (2005), 529–538.

[6] D. H. Li and M. Fukushima, A modified BFGS method and its global convergence in nonconvex minimization, *Journal of Computational and Applied Mathematics*, **129** (2001), 15–35.

[7] A. Miele and J. W. Cantrell, Study on a memory gradient method for the minimization of functions, *Journal of Optimization Theory and Applications*, **3** (1969), 459–470.

[8] J. J. Moré, B. S. Garbow, and K. E. Hillstrom Testing unconstrained optimization software, *ACM Transactions on Mathematical Software*, **7** (1981), 17-41.

[9] J.Nocedal and S.J.Wright, Numerical Optimization, Springer Series in Operations Research, Springer Verlag, New York, 1999.

[10] Y. Narushima, A Nonmonotone Memory Gradient Method for Unconstrained Optimization, *Journal of the Operations Research Society of Japan*, to appear. (See also *Optimization – Modeling and Algorithms –*, The Institute of Statistical Mathematics Cooperative Research Report, **19** (2006), 374–388.)

[11] Y.Narushima and H.Yabe, Global convergence of a memory gradient method for unconstrained optimization, *Computational Optimization and Applications*, **35** (2006), 325–346.

[12] J. Sun and J. Zhang, Global convergence of conjugate gradient methods without line search, *Annals of Operations Research*, **103** (2001), 161–173.

[13] Matrix Market, http://math.nist.gov/MatrixMarket/

Yasushi Narushima
Department of Mathematical Information Science, Tokyo University of Science
1-3, Kagurazaka, Shinjuku-ku, Tokyo, 162-8601, Japan