Journal of the Operations Research
Society of Japan
2007, Vol. 50, No. 1, 31-45

A NONMONOTONE MEMORY GRADIENT METHOD FOR UNCONSTRAINED OPTIMIZATION

Yasushi Narushima
Tokyo University of Science

(Received July 11, 2005; Revised October 5, 2006)

Abstract Memory gradient methods are used for unconstrained optimization, especially large scale problems. They were first proposed by Miele and Cantrell (1969) and 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. In this paper, we propose a nonmonotone memory gradient method based on this work. We show that our method converges globally to the solution. Our numerical results show that the proposed method is efficient for some standard test problems if we choose a parameter included in the method suitably.

Keywords: Nonlinear programming, optimization, memory gradient method, nonmonotone line search, global convergence, large scale problems.

1. Introduction

We consider the following unconstrained optimization problem

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

where $f : \mathbf{R}^n \to \mathbf{R}$ is smooth and its gradient $g(x) \equiv \nabla f(x)$ is available. We denote $g_k \equiv g(x_k)$ and $f_k \equiv f(x_k)$ for simplicity. For solving this problem, iterative methods are widely used. These take the form:

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

where $x_k \in \mathbf{R}^n$ is the k-th approximation to the solution, $\alpha_k > 0$ is a step size and $d_k \in \mathbf{R}^n$ is a search direction. In outline form, the algorithm for a general iterative method is as follows:

Algorithm 1.1 (Iterative Method)
Step 0. Given $x_0 \in \mathbf{R}^n$ and $d_0 \in \mathbf{R}^n$. Set $k = 0$. Go to Step 2.
Step 1. Compute d_k.
Step 2. Compute α_k by using a line search.
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.

There exist many kinds of iterative methods. In general, the Newton method and quasi-Newton methods are very effective in solving the problem (1.1). These methods, however, must hold and manipulate matrices of size $n \times n$. Thus these methods cannot always be applied to large-scale problems. 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 in this class.

The memory gradient method also aims to accelerate the steepest descent method and it was first proposed by Miele and Cantrell [7] and by Cragg and Levy [1]. The search direction of this method is defined by

$$d_k = -\gamma_k g_k + \frac{1}{m}\sum_{i=1}^{m} \beta_{ki} d_{k-i}, \qquad (k \geq 1) \tag{1.3}$$

where $\beta_{ki} \in \mathbf{R}$ $(i = 1, \ldots, m)$ and $\gamma_k \in \mathbf{R}$ are parameters, $\underline{\gamma} \leq \gamma_k < \hat{\gamma}$, and $\underline{\gamma}$ and $\hat{\gamma}$ are given positive constants. The search direction at the first iteration is chosen as the steepest descent direction with a sizing parameter $\gamma_0 > 0$: $d_0 = -\gamma_0 g_0$.

A new memory gradient method has been proposed by Narushima and Yabe [9]. This method always satisfies the sufficient descent condition and converges globally if the Wolfe conditions (see, for example, [10]) are satisfied within the line search strategy. Note that the parameters used in [9] are different from those given by Miele et al.

The technique of the nonmonotone line search was first proposed by Grippo et al. [4]. There are many successful applications or extensions which use nonmonotone line search methods in both unconstrained and constrained optimization. For example, it is applied to Newton type methods by Grippo et al. [4–6] and to the conjugate gradient method by Dai [2]. Moreover the basic analysis of the nonmonotone line search strategy is given by Dai [3].

Now we introduce an algorithm for a nonmonotone line search strategy at the k-th iteration. Let $0 < \lambda_1 \leq \lambda_2$, $0 < \lambda_3 \leq \lambda_4 < 1$, $\delta \in (0, 1)$ and let $\bar{M}$ be a positive integer. Further, let $\alpha_k^{(0)} \in [\lambda_1,\ \lambda_2]$ be an initial trial step size at the k-th iteration. We choose $M(k)$ such that $M(0) = 0$ and $0 \leq M(k) \leq \min\{M(k-1) + 1, \bar{M}\}$ $(k \geq 1)$.

Algorithm 1.2 (Nonmonotone Line Search Strategy)

Step 0. Given $\alpha_k^{(0)}$ *and* $M(k)$. *Set* $i = 0$.

Step 1. If

$$f(x_k + \alpha_k^{(i)} d_k) \leq \max_{0 \leq j \leq M(k)} \{f_{k-j}\} + \delta \alpha_k^{(i)} g_k^T d_k \tag{1.4}$$

holds, set $\alpha_k \equiv \alpha_k^{(i)}$ *and stop. Otherwise go to Step 2.*

Step 2. Choose $\sigma_k^{(i)} \in [\lambda_3,\ \lambda_4]$ *and compute* $\alpha_k^{(i+1)}$ *such that*

$$\alpha_k^{(i+1)} = \alpha_k^{(i)} \sigma_k^{(i)}. \tag{1.5}$$

Step 3. Set $i = i + 1$ *and go to Step 1.*

In Algorithm 1.2, if we always choose 0.5 as $\sigma_k^{(i)}$, then we obtain the bisection nonmonotone line search method. On the other hand, if we set

$$\sigma_k^{(i)} = \max\left\{\lambda_3,\ \min\left\{\lambda_4,\ \frac{0.5\alpha_k^{(i)} g_k^T d_k}{f_k + \alpha_k^{(i)} g_k^T d_k - f(x_k + \alpha_k^{(i)} d_k)}\right\}\right\},$$

then we obtain the quadratic interpolation nonmonotone line search method. Moreover if $\bar{M} = 0$, the above nonmonotone line search reduces to the Armijo line search (see, for instance, [10]). In the nonmonotone line search strategy, the choice of α_k does not force a

monotone decrease of the objective function. However α_k is chosen such that the current objective function value is less than the maximum of the objective function value for the past $M(k)$ iterations.

In this paper, we will consider a nonmonotone memory gradient algorithm which uses the nonmonotone line search strategy. Our algorithm is based on the memory gradient method proposed by Narushima and Yabe [9].

This paper is organized as follows. Our nonmonotone memory gradient algorithm is proposed in the next section. In Section 3, we give a global convergence property of the algorithm. Moreover, we investigate the relation between our method and the R-linear convergence result, under appropriate assumptions. Our numerical results are presented in Section 4, and conclusions are drawn in the last section. Throughout this paper, $\|\cdot\|$ denote the l_2 vector norm.

2. A New Nonmonotone Memory Gradient Method

In this section, we propose a nonmonotone memory gradient method which always satisfies the sufficient descent condition, i.e.,

$$g_k^T d_k \leq -c_1 \|g_k\|^2 \quad \text{for all} \quad k \geq 1 \tag{2.1}$$

for some positive constant c_1.

As in the method given in [9], we define β_{ki} as follows

$$\beta_{ki} = \|g_k\|^2 \psi_{ki}^{\dagger}, \tag{2.2}$$

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} are parameters which satisfy the following condition:

$$\begin{cases} g_k^T d_{k-1} + \|g_k\|\|d_{k-1}\| & < \gamma_k \psi_{k1} \qquad (i = 1), \\ g_k^T d_{k-i} + \|g_k\|\|d_{k-i}\| & \leq \gamma_k \psi_{ki} \qquad (i = 2, \ldots, m). \end{cases} \tag{2.3}$$

Recall that γ_k is a sizing parameter which satisfies $\underline{\gamma} \leq \gamma_k < \hat{\gamma}$ ($\underline{\gamma} > 0$). This choice guarantees $\beta_{k1} > 0$ and $\beta_{ki} \geq 0$ ($i = 2, \ldots, m$), if $\|g_k\| \neq 0$. We note that, if there exists at least one index $i > 1$ such that inequality (2.3) is satisfied as a strict inequality, that is,

$$g_k^T d_{k-i} + \|g_k\|\|d_{k-i}\| < \gamma_k \psi_{ki},$$

then the theorems given below still hold. However, it is natural to use information of the most recent iteration, i.e. $i = 1$. We recall the following result from [9].

Lemma 2.1 *Let d_k be defined by the memory gradient method* (1.3). *We choose β_{ki} and ψ_{ki} that satisfy* (2.2) *and* (2.3) *for all k. Then the search direction* (1.3) *satisfies the sufficient descent condition* (2.1) *for all k.*

Now we present the algorithm of our nonmonotone memory gradient method.

Algorithm 2.1 (Nonmonotone Memory Gradient Method)

Step 0. Given $x_0 \in \mathbf{R}^n$, $\gamma_0 \geq \underline{\gamma}$, $\bar{M}$ and m. Set $d_0 = -\gamma_0 g_0$ and $k = 0$. Go to Step 2.

Step 1. Compute $\gamma_k \geq \underline{\gamma}$ and ψ_{ki} satisfying (2.3), *define β_{ki} by* (2.2) *and generate d_k by* (1.3).

Step 2. Compute α_k by the nonmonotone line search (Algorithm 1.2).

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

Step 4. Set k=k+1 and go to Step 1.

3. Convergence Analysis

In this section, we show the global convergence property of the present method. For this purpose, we make the following standard assumptions.

Assumption 3.1
(A1) *f is bounded below on $\mathbf{R}^n$ and is continuously differentiable in a 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 gradient $g(x)$ is Lipschitz continuous in $\mathcal{N}$, namely, there exists a positive constant L such that*

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

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

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 since f is a continuous function defined on $\mathbf{R}^n$.

In the rest of this section, we assume $g_k \neq 0$ for all k (otherwise a stationary point has been found). The next lemma implies that the angle between the search direction of our method and the steepest descent direction is an acute angle and is bounded away from 90°.

Lemma 3.1 *Let d_k be defined by the memory gradient method* (1.3)*. If we choose ψ_{ki} and β_{ki} that satisfy* (2.3) *and* (2.2) *for all k, there exists a positive constant c_2 such that*

$$\frac{|g_k^T d_k|}{\|g_k\|\|d_k\|} \geq c_2, \tag{3.1}$$

for all k.

Proof. From (1.3), we have

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

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

$$\begin{aligned}
\frac{\|d_k\|^2}{(g_k^T d_k)^2} &= \frac{\|\frac{1}{m}\sum_{i=1}^{m} \beta_{ki} d_{k-i}\|^2}{(g_k^T d_k)^2} - 2\gamma_k \frac{g_k^T d_k}{(g_k^T d_k)^2} - \gamma_k^2 \frac{\|g_k\|^2}{(g_k^T d_k)^2} \\
&= \frac{\|\frac{1}{m}\sum_{i=1}^{m} \beta_{ki} d_{k-i}\|^2}{(g_k^T d_k)^2} - \gamma_k \frac{2}{(g_k^T d_k)} - \gamma_k^2 \frac{\|g_k\|^2}{(g_k^T d_k)^2} \\
&= \frac{\|\frac{1}{m}\sum_{i=1}^{m} \beta_{ki} d_{k-i}\|^2}{(g_k^T d_k)^2} - \left(\frac{1}{\|g_k\|} + \gamma_k \frac{\|g_k\|}{g_k^T d_k}\right)^2 + \frac{1}{\|g_k\|^2} \\
&\leq \frac{\|\frac{1}{m}\sum_{i=1}^{m} \beta_{ki} d_{k-i}\|^2}{(g_k^T d_k)^2} + \frac{1}{\|g_k\|^2} \\
&\leq \left(\frac{\frac{1}{m}\sum_{i=1}^{m} \beta_{ki}\|d_{k-i}\|}{|g_k^T d_k|}\right)^2 + \frac{1}{\|g_k\|^2}.
\end{aligned} \tag{3.2}$$

On the other hand, we obtain from Lemma 2.1, (1.3), (2.2) and (2.3)

$$\begin{aligned} |g_k^T d_k| &= -g_k^T d_k \\ &= \gamma_k \|g_k\|^2 - \frac{1}{m}\sum_{i=1}^{m} \beta_{ki} g_k^T d_{k-i} \\ &= \frac{1}{m}\sum_{i=1}^{m} (\gamma_k \|g_k\|^2 - \beta_{ki} g_k^T d_{k-i}) \\ &\geq \frac{1}{m}\sum_{i=1}^{m} (\gamma_k \psi_{ki} - g_k^T d_{k-i})\beta_{ki} > 0. \end{aligned} \tag{3.3}$$

The first equality follows from the fact that $g_k^T d_k < 0$ for all k which is established by Lemma 2.1, and the last inequality follows from (2.3) and $\beta_{k1} > 0$. Noting that $\gamma_k \psi_{ki} \geq g_k^T d_{k-i} + \|g_k\|\|d_{k-i}\|$ and multiplying this by $\beta_{ki} \geq 0$, we have

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

Summing the above inequality, we obtain

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

Therefore inequality (3.3) yields

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

Finally it follows from (3.2) and (3.4) that

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

This implies that (3.1) holds with $c_2 = \frac{1}{\sqrt{2}}$. □

By using Lemma 2.1 and Lemma 3.1, we have the following convergence theorem.

Theorem 3.1 *Suppose that Assumption* 3.1 *holds. Let the sequence* $\{x_k\}$ *be generated by Algorithm* 2.1. *Then our method converges in the sense that*

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

Proof. Define $l(k)$ to be a number such that

$$k - M(k) \leq l(k) \leq k \quad \text{and} \quad f_{l(k)} = \max_{0\leq j\leq M(k)} \{f_{k-j}\}$$

for every k. By (1.4) and the fact that $M(k+1) \leq M(k) + 1$, we obtain

$$\begin{aligned} f_{l(k+1)} &= \max_{0\leq j\leq M(k+1)} \{f_{k+1-j}\} \\ &\leq \max_{0\leq j\leq M(k)+1} \{f_{k+1-j}\} \\ &= \max\{f_{l(k)},\ f_{k+1}\} \\ &= f_{l(k)}. \end{aligned}$$

Therefore we see that the sequence $\{f_{l(k)}\}$ is non-increasing. Moreover, by (1.4), we have (for $k > \bar{M}$)

$$\begin{aligned} f_{l(k+1)} &\leq f_{l(l(k))} \\ &\leq \max_{0\leq j\leq M(l(k)-1)}\{f_{l(k)-1-j}\} + \delta\alpha_{l(k)-1}g_{l(k)-1}^T d_{l(k)-1} \\ &= f_{l(l(k)-1)} + \delta\alpha_{l(k)-1}g_{l(k)-1}^T d_{l(k)-1}. \end{aligned}$$

From Assumption 3.1 and the fact that the sequence $\{f_{l(k)}\}$ is non-increasing, $\{f_{l(k)}\}$ has a limit. In the remainder of the proof, we replace the subsequence $\{l(k)-1\}$ by $\{k'\}$. Therefore

$$\lim_{k'\to\infty} \alpha_{k'} g_{k'}^T d_{k'} = 0 \tag{3.5}$$

holds.

If the theorem is not true, there exists a constant $c_3 > 0$ such that

$$\|g_k\| \geq c_3 \tag{3.6}$$

for all k. From Lemma 2.1 and (3.6), we have

$$\alpha_{k'} g_{k'}^T d_{k'} \leq -c_1\alpha_{k'}\|g_{k'}\|^2 \leq -\alpha_{k'}c_1c_3^2 < 0. \tag{3.7}$$

It follows from (3.5) and (3.7) that

$$\lim_{k'\to\infty} \alpha_{k'} = 0.$$

This equation implies that when k' is sufficiently large, $\alpha_{k'}^{(0)} (> \lambda_1)$ does not satisfy (1.4) i.e. $\alpha_{k'} = \alpha_{k'}^{(i)}$ holds for some $i \neq 0$. Therefore from (1.4) we have

$$\begin{aligned} f(x_{k'} + \alpha_{k'}^{(i-1)} d_{k'}) &> \max_{0\leq j\leq M(k')}\{f_{k'-j}\} + \delta\alpha_{k'}^{(i-1)} g_{k'}^T d_{k'} \\ &\geq f_{k'} + \delta\alpha_{k'}^{(i-1)} g_{k'}^T d_{k'}. \end{aligned} \tag{3.8}$$

By the mean value theorem and the Lipschitz continuity of g, we obtain

$$\begin{aligned} f(x_{k'} + \alpha_{k'}^{(i-1)} d_{k'}) - f_{k'} &= \alpha_{k'}^{(i-1)} g(x_{k'} + \tau\alpha_{k'}^{(i-1)} d_{k'})^T d_{k'} \\ &= \alpha_{k'}^{(i-1)} \left[g_{k'}^T d_{k'} + \{g(x_{k'} + \tau\alpha_{k'}^{(i-1)} d_{k'}) - g_{k'}\}^T d_{k'} \right] \\ &\leq \alpha_{k'}^{(i-1)} \{ g_{k'}^T d_{k'} + L\tau\alpha_{k'}^{(i-1)} \|d_{k'}\|^2 \} \\ &= \alpha_{k'}^{(i-1)} g_{k'}^T d_{k'} + L\tau(\alpha_{k'}^{(i-1)}\|d_{k'}\|)^2, \end{aligned}$$

for some constant τ such that $0 < \tau < 1$. It follows from (1.5) and (3.8) that

$$g_{k'}^T d_{k'} + L\tau\frac{\alpha_{k'}}{\sigma_{k'}^{(i-1)}}\|d_{k'}\|^2 > \delta g_{k'}^T d_{k'}.$$

Taking the conditions $\delta \in (0,\ 1)$ and $\lambda_3 \leq \sigma_{k'}^{(i-1)}$ into account, we can write

$$\alpha_{k'} > \bar{c}\frac{|g_{k'}^T d_{k'}|}{\|d_{k'}\|^2}, \tag{3.9}$$

where $\bar{c} = \frac{\lambda_3(1-\delta)}{L\tau} > 0$. Since (3.9) yields

$$\alpha_{k'}|g_{k'}^T d_{k'}| > \bar{c}\frac{|g_{k'}^T d_{k'}|^2}{\|d_{k'}\|^2} > 0,$$

we have, from (3.5),

$$\lim_{k'\to\infty} \frac{|g_{k'}^T d_{k'}|^2}{\|d_{k'}\|^2} = 0. \tag{3.10}$$

On the other hand, it follows from Lemma 3.1 and (3.6) that

$$\frac{|g_{k'}^T d_{k'}|}{\|d_{k'}\|} \geq c_2\|g_{k'}\| \geq c_2 c_3 > 0.$$

This contradicts (3.10). Therefore the proof is complete. □

This theorem implies that, if we choose γ_k and ψ_{ki} $(i = 1, \ldots, m)$ to satisfy the condition (2.3), then global convergence of our method is achieved. Based on this theorem, we can propose several kinds of search directions.

Since Algorithm 1.2 reduces to the Armijo line search for $\bar{M} \equiv 0$, we directly obtain the next corollary from Theorem 3.1 without proof.

Corollary 3.1 *Suppose that Assumption* 3.1 *holds. Let the sequence* $\{x_k\}$ *be generated by the memory gradient method with the Armijo line search strategy, i.e. Algorithm* 2.1 *with* $\bar{M} \equiv 0$. *Then our method converges in the sense that*

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

In [9], the global convergence property of our memory gradient method with the Wolfe conditions has been established. On the other hand, this corollary implies that we can prove the convergence property with a weaker condition than the Wolfe conditions.

In the rest of this section, we study strong results for the restricted version of Algorithm 2.1. To establish strong properties, we require ψ_{ki} to satisfy

$$\max\left\{g_k^T d_{k-i},\ \nu\|g_k\|\|d_{k-i}\|\right\} + \|g_k\|\|d_{k-i}\| \leq \psi_{ki}\gamma_k \quad (i = 1, \ldots, m), \tag{3.11}$$

where $\nu > -1$ is a constant we choose in the algorithm. Note that if we choose ψ_{ki} which satisfy (3.11), then these also satisfy (2.3) and ψ_{ki} does not become zero (whenever $g_k \neq 0$). The following lemma is obtained.

Lemma 3.2 *Let* d_k *be defined by the memory gradient method* (1.3). *If we choose* ψ_{ki} *and* β_{ki} *that satisfy* (3.11) *and* (2.2) *for all* k, *then there exists a positive constant* c_4 *such that*

$$\|d_k\| \leq c_4\|g_k\| \tag{3.12}$$

for all k.

Proof. It follows from (1.3), (2.2), (3.11) and $\psi_{ki} \neq 0$ $(i = 1, \cdots, m)$ that

$$\begin{aligned}
\|d_k\| &= \left\| -\gamma_k g_k + \frac{1}{m}\sum_{i=1}^{m} \|g_k\|^2 \psi_{ki}^{\dagger} d_{k-i} \right\| \\
&\leq \hat{\gamma}\|g_k\| + \frac{1}{m}\sum_{i=1}^{m} \frac{\|d_{k-i}\|\|g_k\|^2}{\psi_{ki}} \\
&\leq \hat{\gamma}\|g_k\| + \frac{1}{m}\sum_{i=1}^{m} \frac{\hat{\gamma}\|d_{k-i}\|\|g_k\|^2}{\max\{g_k^T d_{k-i},\ \nu\|g_k\|\|d_{k-i}\|\} + \|g_k\|\|d_{k-i}\|} \\
&\leq \hat{\gamma}\|g_k\| + \frac{1}{m}\sum_{i=1}^{m} \frac{\hat{\gamma}\|g_k\|}{1+\nu} \\
&= \hat{\gamma}\left(1 + \frac{1}{1+\nu}\right)\|g_k\|.
\end{aligned}$$

Therefore the proof is complete with $c_4 = \hat{\gamma}\left(\frac{2+\nu}{1+\nu}\right)$. □

By using this lemma, the following two desired properties are easily obtained.

First, we derive a strong convergence result for the restricted version of our algorithm. The following lemma was proved by Dai [3]. A similar proof can be given for this case, although there are a few differences between the situation that Dai [3] considers and our nonmonotone line search algorithm.

Lemma 3.3 *Suppose that Assumption* 3.1 *holds. Consider any iterative method* (1.2) *in which d_k satisfies* (2.1) *and* (3.12)*, and in which α_k is obtained by Algorithm* 1.2*. Then there exists a positive constant c_5 such that*

$$\|g_{k+1}\| \leq c_5\|g_k\| \tag{3.13}$$

for all k. Further, we have that

$$\lim_{k\to\infty} \|g_k\| = 0. \tag{3.14}$$

From this lemma, we obtain the following theorem.

Theorem 3.2 *Suppose that Assumption* 3.1 *holds. Let the sequence $\{x_k\}$ be generated by Algorithm* 2.1*. Further we choose ψ_{ki} and γ_k which satisfy* (3.11)*. Then our method converges in the sense that*

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

Proof. By Lemmas 2.1, 3.2 and 3.3, we obtain the result immediately.

□

Next, we investigate the convergence rate of our method for uniformly convex functions. For this purpose, we make the following assumption.

Assumption 3.2

(A3) *The objective function is a uniformly convex function, namely, there exist positive constants η_1 and η_2 such that*

$$\eta_1\|x-y\|^2 \leq (x-y)^T[g(x)-g(y)] \leq \eta_2\|x-y\|^2$$

for any $x,\ y \in \mathbf{R}^n$

Under this assumption, Dai [3] proved the following lemma.

Lemma 3.4 *Suppose that Assumption* 3.2 *holds and that the objective function* $f(x)$ *is sufficiently smooth. Consider any iterative method* (1.2) *in which* d_k *satisfies* (2.1) *and* (3.12)*, and in which* α_k *is obtained by Algorithm* 1.2*. Then there exist constants* $c_6 > 0$ *and* $c_7 \in (0,\ 1)$ *such that*

$$f(x_k) - f(x^*) \leq c_6 c_7^{k+1} \left[f(x_0) - f(x^*) \right],$$

where x^* *is a unique minimizer of* f*.*

This lemma implies that the convergence rate is R-linear. By using this lemma, we obtain the following theorem.

Theorem 3.3 *Suppose that Assumption* 3.2 *holds and that the objective function* $f(x)$ *is sufficiently smooth. Let the sequence* $\{x_k\}$ *be generated by Algorithm* 2.1*. Further, we assume that* ψ_{ki} *and* γ_k *are chosen to satisfy* (3.11)*. Then the sequence* $\{x_k\}$ *converges R-linearly to the solution* x^**.*

Proof. By Lemmas 2.1, 3.2 and 3.4, the results follows immediately.

□

4. Numerical Results

In this section, we report some preliminary numerical results of Algorithm 2.1. We denote $s_{k-1} = x_k - x_{k-1}$ and $y_{k-1} = g_k - g_{k-1}$.

In our experiment, we first chose γ_k and next determined ψ_{ki} $(i = 1, \ldots, m)$ that satisfied condition (3.11). We choose $\gamma_0 = 1$ and

$$\gamma_k = \begin{cases} 1 & \text{if } \dfrac{z_{k-1}^T s_{k-1}}{z_{k-1}^T z_{k-1}} < 10^{-15} \\ \dfrac{z_{k-1}^T s_{k-1}}{z_{k-1}^T z_{k-1}} & \text{otherwise,} \end{cases} \tag{4.1}$$

where

$$z_{k-1} = y_{k-1} + \frac{\theta_{k-1}}{s_{k-1}^T u_{k-1}} u_{k-1},$$

u_{k-1} is any vector such that $s_{k-1}^T u_{k-1} \neq 0$ and

$$\theta_{k-1} = 6(f(x_{k-1}) - f(x_k)) + 3(g_{k-1} + g_k)^T s_{k-1}.$$

This choice of the sizing parameter was proposed in [9]. In the numerical experiments, we chose $u_{k-1} = s_{k-1}$. For a given γ_k, we used ψ_{ki} $(i = 1, \ldots, m)$ defined by

$$\psi_{ki} = \frac{\max\left\{ g_k^T d_{k-i},\ \nu \|g_k\| \|d_{k-i}\| \right\} + \|g_k\| \|d_{k-i}\| + n}{\gamma_k} \quad (i = 1, \ldots, m), \tag{4.2}$$

where $\nu = -0.8$. Note that this choice of ψ_{ki} satisfies condition (3.11).

In the nonmonotone line search strategy, the initial step size $\alpha_k^{(0)} = 1$ was always chosen, and $\sigma_k^{(i)} = 0.5$ in all cases. We set the other parameters as follows: $\delta = 10^{-4}$ and

$$M(k) = \begin{cases} k & k < \bar{M} \\ \bar{M} & \text{otherwise.} \end{cases}$$

The stopping condition was

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

We tested our method with the values $m = 0,\ 1,\ 3,\ 5,\ 7,\ 9$ and $\bar{M} = 0,\ 1,\ 3,\ 5,\ 7,\ 9$. The choice $m = 0$ yields the steepest descent method with the sizing parameter (4.1) (denoted by S-SD), namely, $d_k = -\gamma_k g_k$. Moreover, if we choose $\bar{M} = 0$, then Algorithm 1.2 reduces to the Armijo line search method.

In order to compare our method with other methods, we used the limited memory BFGS quasi-Newton method (denoted by LQN) with memory $\hat{m} = 3,\ 5,\ 7$ (see [10] for example). In LQN, the step size α_k which satisfies the Armijo condition was chosen in the line search and an initial Hessian approximation was set to $(y_{k-1}^T s_{k-1}/y_{k-1}^T y_{k-1})I$, where I is the unit matrix. Moreover, in LQN, if the search direction does not generate a descent direction, then we use the steepest descent direction.

Table 1: Test problems

Name	Dimension
Extended Rosenbrock Function	n=10000 or 100000
Extended Powell Singular Function	n=10000 or 100000
Trigonometric Function	n=10000 or 100000
Broyden Tridiagonal Function	n=10000 or 100000
Wood Function	n=4

The test problems we used are described in Moré et al. [8]. In Table 1, the first column and the second column denote the problem name and the dimension of the problem, respectively. Since we are interested in the performance for the large scale problems, we list only large problems although we tested other small problems in Moré et al. [8]. We present the results for the Wood Function ($n = 4$), because this function is singular at the solution. Tables 2 to 11 give the numerical results of the experiments in the general form: (number of iterations)/(number of function value evaluations). We write "*Failed* " when the number of iterations exceeded 1000. In Tables 2 to 10, we list the numerical results of our method, where the column '$m = 0$' corresponds to the numerical results for S-SD. In each table, the results printed in boldface imply that the nonmonotone memory gradient algorithm performed better than the *monotone* memory gradient algorithm ($\bar{M} = 0$). In Table 11, we list the numerical results for the LQN method.

From now on, for simplicity, MG and NMG will denote the monotone memory gradient algorithm and the nonmonotone memory gradient algorithm, respectively. Comparing MG and NMG in each column for the Extended Rosenbrock Function and the Extended Powell Singular Function in Tables 2 to 5, we can see that NMG performed better, depending on the parameters m and $\bar{M}$. In particular, the number of function evaluations for NMG was much fewer than for MG. For the Trigonometric Function (Tables 6 and 7), we do not observe any significant improvement for NMG. In this case, we see that NMG is merely comparable with MG. For the Broyden Tridiagonal Function with $n = 10000$ in Table 8, we find that there are good results for NMG, for instance, $m = 5,\ \bar{M} = 1$ and the column corresponding to $m = 1$. However we observe that NMG performed poorly in the cases $m = 9,\ \bar{M} = 5,\ 7,\ 9$. For the Broyden Tridiagonal Function with $n = 100000$ in Table 9, we can see that NMG is comparable with MG except for the column $m = 3$. From Tables 2 to 11, we see that our methods are comparable with S-SD. In Tables 2 to 5 and 8 to 10, we see that if we choose suitable values for the parameters, NMG outperforms S-SD. Comparing

our method with LQN, LQN outperforms our method for the Powell Singular Function and the Wood Function.

The performance of our method depends on the choice of parameters m and $\bar{M}$, and we have not yet found the best choices. Even if we choose large values for m and $\bar{M}$, these are not necessarily the best (for instance, in Table 2, $\bar{M} = 9$ is good, while in Table 9, $\bar{M} = 0$ is good). Though it may be difficult to propose the best choice theoretically, the choices $m = 5,\ 7$ and $\bar{m} = 7,\ 9$ are better in our experiments. In addition, we obtain $d_k \approx -\gamma_k g_k$ if n is extremely large and g_k is small (because $\beta_{ki} \leq \gamma_k \|g_k\|/n$ holds). It therefore follows that, in such situations, our method may tend to exhibit slow convergence, like the method of steepest descent. Thus we may need further study about choices for the parameter ψ_{ki}.

Finally, we present Figure 1, as an example, to demonstrate the local behavior of our method. This figure gives the values of $\log_{10}[f(x)]$ for the Extended Rosenbrock Function, where $f(x)$ becomes zero at the optimal solution. In the figure, the triangle symbols and the diamond symbols show the behavior of the function value for $m = 7, \bar{M} = 0$ (monotone case) and $m = 7, \bar{M} = 9$ (nonmonotone case), respectively. We are unable to observe any strong indication of superlinear convergence here.

Table 2: Extended Rosenbrock Function ($n = 10000$)

$\bar{M}$ \ m	0	1	3	5	7	9
0	63/123	73/131	67/118	74/130	72/127	69/133
1	66/112	80/117	72/109	**72/109**	**65/103**	70/106
3	**61/95**	80/117	76/112	**72/109**	**66/101**	73/109
5	**59/81**	75/93	77/99	**63/87**	**64/86**	70/97
7	**59/81**	76/98	**67/88**	**64/85**	**64/86**	**68/92**
9	**59/80**	**68/81**	**65/80**	**57/72**	**47/63**	**67/88**

Table 3: Extended Rosenbrock Function ($n = 100000$)

$\bar{M}$ \ m	0	1	3	5	7	9
0	63/123	76/136	67/118	76/132	72/127	71/135
1	68/114	80/117	72/109	**75/112**	**66/104**	**70/106**
3	**61/95**	80/117	76/112	**75/112**	**66/101**	73/109
5	**59/81**	**75/93**	77/100	**63/87**	**67/89**	**70/97**
7	**59/81**	**76/98**	**67/88**	**64/85**	**67/89**	**68/92**
9	**59/80**	**68/81**	**65/80**	**60/75**	**48/64**	**70/91**

Table 4: Extended Powell Singular Function ($n = 10000$)

$\bar{M}$ \ m	0	1	3	5	7	9
0	239/405	245/408	191/301	310/484	313/476	225/384
1	**172/233**	**226/293**	214/272	**253/347**	**191/255**	**205/268**
3	**187/269**	**217/260**	262/336	**197/267**	**186/254**	**190/257**
5	**182/241**	**237/301**	212/269	**197/267**	**186/254**	**211/277**
7	**186/257**	**195/246**	**164/213**	**213/279**	**176/236**	**206/252**
9	**153/179**	**196/254**	**145/166**	**156/194**	**198/241**	**204/235**

Table 5: Extended Powell Singular Function ($n = 100000$)

$\bar{M}$ \ m	0	1	3	5	7	9
0	212/361	237/414	233/382	300/482	243/388	258/426
1	223/315	**215/274**	236/318	**247/350**	251/338	286/385
3	293/414	267/341	**265/345**	**246/321**	**219/280**	**223/311**
5	**204/273**	267/341	**223/294**	**246/321**	**219/280**	**184/247**
7	221/310	**205/266**	**198/269**	**227/296**	**214/297**	**217/270**
9	**172/217**	251/325	**180/208**	**164/197**	**220/267**	**207/259**

Table 6: Trigonometric Function ($n = 10000$)

$\bar{M}$ \ m	0	1	3	5	7	9
0	59/61	61/62	65/69	64/66	61/65	66/69
1	**59/60**	**61/62**	66/68	**60/61**	63/65	**65/66**
3	**59/60**	**61/62**	66/67	**60/61**	63/65	**65/66**
5	**59/60**	**61/62**	66/67	**60/61**	63/65	**65/66**
7	**59/60**	**61/62**	66/67	**60/61**	**60/61**	**65/66**
9	**59/60**	**61/62**	66/67	**60/61**	**60/61**	**65/66**

Table 7: Trigonometric Function ($n = 100000$)

$\bar{M}$ \ m	0	1	3	5	7	9
0	42/43	50/52	42/43	59/60	52/54	45/46
1	**42/43**	**50/51**	**42/43**	**59/60**	**47/48**	**45/46**
3	**42/43**	**50/51**	**42/43**	**59/60**	**47/48**	**45/46**
5	**42/43**	**50/51**	**42/43**	**59/60**	**47/48**	**45/46**
7	**42/43**	**50/51**	**42/43**	**59/60**	**47/48**	**45/46**
9	**42/43**	**50/51**	**42/43**	**59/60**	**47/48**	**45/46**

Table 8: Broyden Tridiagonal Function ($n = 10000$)

$\bar{M}$ \ m	0	1	3	5	7	9
0	85/111	83/104	126/148	95/118	107/132	94/118
1	**82/90**	**81/89**	**120/132**	**73/84**	**90/103**	**90/102**
3	**82/90**	**79/86**	**105/117**	165/174	151/162	100/116
5	**82/90**	**79/86**	**105/117**	165/174	151/162	540/565
7	**82/90**	**79/86**	**91/99**	134/143	151/162	585/608
9	88/96	**79/86**	**91/99**	134/143	151/162	525/549

Table 9: Broyden Tridiagonal Function ($n = 100000$)

$\bar{M}$ \ m	0	1	3	5	7	9
0	54/80	55/75	57/72	51/69	50/66	55/78
1	58/69	77/88	65/78	55/65	57/68	56/67
3	56/64	**54/63**	156/169	54/63	58/67	61/71
5	56/64	56/63	151/165	54/63	58/67	61/70
7	56/64	56/63	151/163	60/68	57/65	59/67
9	56/64	56/63	151/163	60/68	57/65	59/67

Table 10: Wood Function ($n = 4$)

$\bar{M}$ \ m	0	1	3	5	7	9
0	358/461	395/497	332/427	148/223	242/315	336/443
1	**303/351**	**320/368**	661/825	418/472	264/315	**336/390**
3	**282/323**	397/448	616/770	437/485	**192/231**	365/412
5	**250/293**	443/497	516/595	292/336	**141/178**	373/446
7	**272/306**	**356/397**	545/609	369/421	**141/178**	**328/390**
9	**270/303**	**352/391**	451/518	340/392	**141/178**	**332/407**

Table 11: Results of LQN

P	n	LQN $\hat{m} = 3$	LQN $\hat{m} = 5$	LQN $\hat{m} = 7$
Extended Rosenbrock Function	10000	41/84	41/100	31/166
	100000	41/84	41/100	32/167
Extended Powell Singular Function	10000	84/122	57/78	57/76
	100000	155/211	57/78	58/77
Trigonometric Function	10000	44/45	41/43	41/42
	100000	24/26	31/32	23/24
Broyden Tridiagonal Function	10000	68/80	66/74	60/66
	100000	72/83	70/85	61/71
Wood Function	4	40/61	33/53	30/47

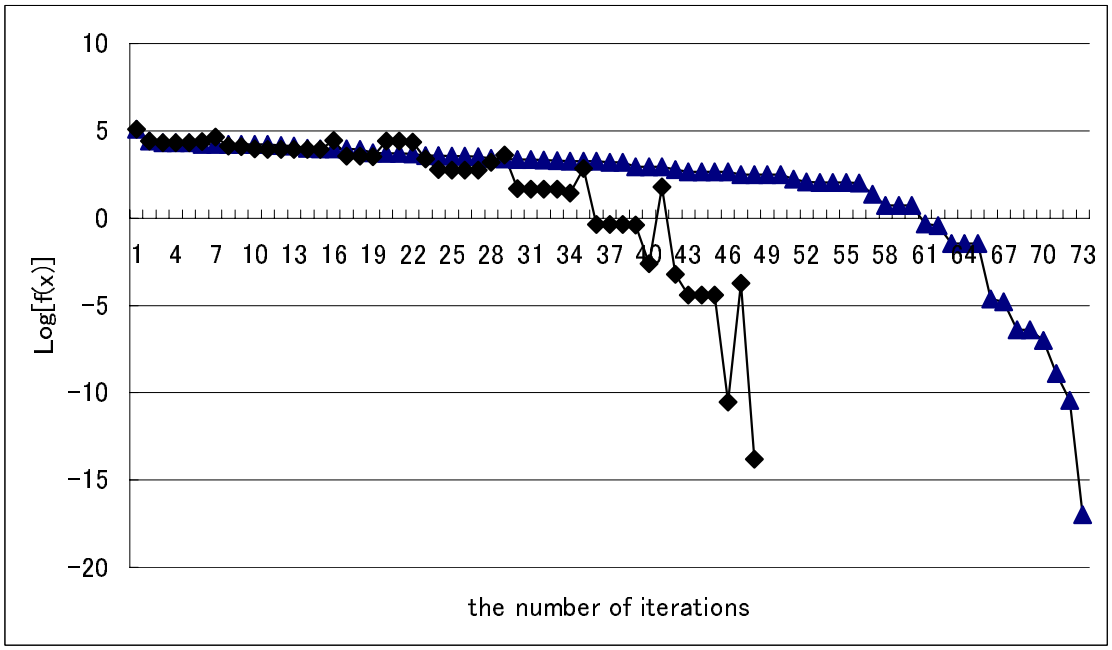

Figure 1. The function value for Extended Rosenbrock Function $n = 10000,\ m = 7,\ \bar{M} = 0$ (symbol $\triangle$) and $m = 7,\ \bar{M} = 9$ (symbol $\diamond$)

5. Conclusion

In this paper, we have proposed a new nonmonotone memory gradient method which always satisfies the sufficient descent condition, and we have proved the global convergence of our method. We have also derived a stronger convergence result for a restricted version of the new method. Finally, we have demonstrated the R-linear convergence of the proposed method in the case where the objective function is uniformly convex.

From the numerical experiments, we see that our method is comparable with S-SD and that the numerical performance of the proposed method depends on the parameters m and $\bar{M}$. We are interested to investigate further new good choices of ψ_{ki} in theory and for practical computation.

Acknowledgements

The author would like to thank the referees for valuable comments. The author is grateful to Prof. Hiroshi Yabe of Tokyo University of Science for his valuable advice and encouragement. The author is grateful to Prof. John A. Ford of University of Essex for his valuable advice. The author was supported in part by a Grant for the Promotion of the Advancement of Education and Research in Graduate Schools of the Ministry of Education, Culture, Sports, Science and Technology of Japan.

References

[1] 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.

[2] Y.H. Dai: A nonmonotone conjugate gradient algorithm for unconstrained optimization. *Journal of System Science and Complexity*, **15** (2002), 139–145.

[3] Y.H. Dai: On the nonmonotone line search. *Journal of Optimization Theory and Applications*, **112** (2002) 315–330.

[4] L. Grippo, F. Lampariello, and S. Lucidi: A nonmonotone line search technique for Newton's method. *SIAM Journal on Numerical Analysis*, **23** (1986), 707–716.

[5] 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.

[6] L. Grippo, F. Lampariello, and S. Lucidi: A class of nonmonotone stabilization methods in unconstrained optimization. *Numeriche Mathematik*, **59** (1991), 779–805.

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

[10] J. Nocedal and S.J. Wright: *Numerical Optimization, Springer Series in Operations Research* (Springer Verlag, New York, 1999).

Yasushi Narushima
Tokyo University of Science
1-3 Kagurazaka, Shinjuku-ku,

Tokyo 162-8601, Japan

