Memory gradient method with Goldstein line search

Zhen-Jun Shi\textsuperscript{a,b,*}, Jie Shen\textsuperscript{b}

\textsuperscript{a} College of Operations Research and Management, Qufu Normal University, Rizhao, Shandong 276826, PR China
\textsuperscript{b} Department of Computer and Information Science, University of Michigan-Dearborn, Michigan, MI48128-1491, USA

Received 23 December 2005; accepted 14 September 2006

Abstract

In this paper, we present a multi-step memory gradient method with Goldstein line search for unconstrained optimization problems and prove its global convergence under some mild conditions. We also prove the linear convergence rate of the new method when the objective function is uniformly convex. Numerical results show that the new algorithm is suitable to solve large-scale optimization problems and is more stable than other similar methods in practical computation.

Keywords: Unconstrained optimization; Memory gradient method; Line search; Convergence

1. Introduction

The method for solving an unconstrained optimization problem

\[ \min f(x), \quad x \in \mathbb{R}^n, \]  

with \( f : \mathbb{R}^n \to \mathbb{R} \) being a continuously differentiable function usually takes the form

\[ x_{k+1} = x_k + \alpha_k d_k, \quad k = 0, 1, 2, \ldots, \]  

at the \( k \)th iteration, where \( d_k \) is a descent direction and \( \alpha_k \) is a step size. Given an initial point \( x_0 \), we can produce a sequence \( \{x_k\} \) by using (2). We hope that the sequence \( \{x_k\} \) can converge to a minimizer of (1) or a stationary point \( x^* \) of (1) for which \( \nabla f(x^*) = 0 \). Assume that \( x_k \) is the current iterate at the \( k \)th iteration, we denote \( f(x_k) \) by \( f_k \), \( \nabla f(x_k) \) by \( g_k \), and \( f(x^*) \) by \( f^* \), respectively.

Generally, if \( d_k = -g_k \) then the related method is called steepest descent method. This method converges very slowly and often yields zigzag phenomenon in practical computation. The conjugate gradient method is a useful technique for solving large-scale optimization problems because it avoids, like the steepest descent method, the...
computation and the storage of some matrices associated with the Hessian of objective functions. It has the form (2) with
\[
d_k = \begin{cases} 
-g_k, & \text{if } k = 0; \\
-g_k + \beta_k d_{k-1}, & \text{if } k \geq 1,
\end{cases}
\]
(3)
where \( \beta_k \) is a parameter that determines different conjugate gradient methods [1–4].

Some multi-step quasi-Newton methods use previous multi-step iterative information to generate a new iterate at each iteration and have the stability property [5–9] in solving ill-conditioned minimization problems. However, they need to memorize and compute some matrices in each iteration. Therefore, like the quasi-Newton method, they are not suitable to solve large-scale minimization problems.

Similar methods to the conjugate gradient method are the memory gradient method or the super-memory gradient method (multi-step gradient method) [10–12]. They avoid the computation and the storage of some matrices associated with Newton-type methods and thus are also suitable to solve large-scale optimization problems. Many memory gradient methods often use exact line search or multi-dimensional search to choose a step size at each iteration and have theoretical global convergence under some mild conditions [13–16]. However, it is difficult or time-consuming to implement an exact line search in practical computation. We should seek a suitable search direction and use some available inexact line searches to choose a step size at each iteration for memory gradient methods and guarantee the global convergence.

As a rule, how to choose an available search direction and a suitable step size at each iteration is the main task in developing memory gradient methods. Some new curve search rules were proposed for guaranteeing the global convergence of memory gradient methods [17,18]. The Wolfe line search and the Armijo line search can guarantee the global convergence in some cases [19,20]. Can the Goldstein line search guarantee the global convergence?

In this paper, we present a new multi-step memory gradient method with Goldstein line search for unconstrained optimization problems and prove its global convergence under some mild conditions. We also investigate the linear convergence rate of this new method when the objective function is uniformly convex. Numerical results show that the new algorithm is suitable to solve large-scale optimization problems and is more stable than other similar methods in practical computation.

The rest of this paper is organized as follows. In Section 2, we describe the algorithm and analyze its simple properties. In Sections 3 and 4 we prove its global convergence under some mild conditions. Linear convergence rate is analyzed in Section 5. Numerical results and comparisons are given in Section 6.

2. New algorithm

In this section, we first introduce the algorithm and then give some properties of this algorithm. Once the search direction \( d_k \) has been determined, there are several rules for choosing step size \( \alpha_k \) for (2). We use Goldstein line search [21] in this new algorithm.

**Goldstein line search.** At the \( k \)th iteration, \( \alpha_k \) is chosen to satisfy
\[
\alpha_k \mu_2 g_k^T d_k \leq f(x_k + \alpha_k d_k) - f_k \leq \alpha_k \mu_1 g_k^T d_k,
\]
where \( 0 < \mu_1 < \mu_2 < 1 \).

**Algorithm A.** *Step 0.* Set some parameters \( 0 < \rho < 1, 0 < \mu_1 < \mu_2 < 1 \), fix an integer \( m \geq 2 \), choose \( x_0 \in \mathbb{R}^n \) and set \( k := 0; \)

*Step 1.* If \( \|g_k\| = 0 \) then stop, else go to step 2;

*Step 2.* \( x_{k+1} = x_k + \alpha_k d_k(\beta_k^{(k)}, \ldots, \beta_k^{(k)}) \), where
\[
d_k(\beta_k^{(k)}, \ldots, \beta_k^{(k)}) = \begin{cases} 
-g_k, & \text{if } k \leq m - 1; \\
-\sum_{i=1}^{m} \beta_k^{(k)} g_{k-i+1}, & \text{if } k \geq m,
\end{cases}
\]
(5)
in which
\[
\beta_k^{(k)} = 1 - \sum_{i=2}^{m} |\beta_k^{(k)}_{i-1}|, \quad \beta_k^{(k)}_{i-1} = \begin{cases} s_k^i, & \text{if } g_k^T g_{k-1}^{i+1} \geq 0; \\ -s_k^i, & \text{otherwise}, \end{cases}
\]
\[(i = 2, 3, \ldots, m)\]
and
\[
s_k^i = \frac{\rho}{m-1} \frac{\|g_k\|^2}{\|g_k\|^2 + |g_k^T g_{k-1}^{i+1}|}, \quad (i = 2, \ldots, m),
\]
and \(\alpha_k\) is defined by Goldstein line search;

**Step 3.** Set \(k := k + 1\) and go to Step 1.

**Remark.** In the above algorithm, we find a new search direction \(d_k(\beta_k^{(k)}_{k-m+1}, \ldots, \beta_k^{(k)})\) for \(k \geq m\). It is a perturbation of the negative gradient of \(f(x)\) at the iterate \(x_k\). It is also a descent direction of \(f(x)\) at \(x_k\) (see Lemma 2.1 below). For the choice of \(\beta_k^{(k)}_{i-1}\), the motivation is that, the greater the relative size of \(|g_k^T g_{k-1}^{i+1}|\) to \(\|g_k\|^2\), the smaller the weight of \(g_{k-1}^{i+1}\) in the direction \(d_k(\beta_k^{(k)}, \ldots, \beta_k^{(k-m+1)})\).

In fact, it might be quite worthwhile to consider some other possible choices of these parameters. For example, the search direction in the classical conjugate gradient method can be written in the form
\[
d_k = -g_k - \sum_{i=1}^{k} \beta_k^{(k)}_{i} g_{k-i}.
\]
If we truncate some terms the search direction will be changed to
\[
d_k = -g_k - \sum_{i=1}^{m} \beta_k^{(k)}_{i} g_{k-i}, \quad k \geq m.
\]
This is very similar to (5). We can try to find some other parameters \(\beta_k^{(k)}_{i} \mid_{k-j}\) for multi-step conjugate gradient methods [22,23]. However, the global convergence of this kind of methods cannot be guaranteed in many situations. We have to restrict the parameters to a scope to guarantee the global convergence of related methods. In this paper, we find a choice of these parameters to guarantee the global convergence. But the related truncated method is not a conjugate gradient method. It may be called the memory gradient method. We believe that there should be many choices of parameters \(\beta_k^{(k)}_{i} \mid_{k-j}\) to make the memory gradient method converge globally.

For simplicity, we sometimes denote \(d_k(\beta_k^{(k)}_{k-m+1}, \ldots, \beta_k^{(k)})\) by \(d_k\), and \(\| \cdot \|\) denotes the Euclidean norm on \(\mathbb{R}^n\). Algorithm A has the following properties.

**Lemma 2.1.** For all \(k \geq 0\),
\[
g_k^T d_k \leq -(1 - \rho)\|g_k\|^2.
\]

**Proof.** If \(k \leq m - 1\), then
\[
-g_k^T d_k = \|g_k\|^2 \geq (1 - \rho)\|g_k\|^2.
\]
If \(k \geq m\) then, by \(|\beta_k^{(k)}_{k-i+1}| \leq s_k^i, \quad (i = 2, \ldots, m)\), we have
\[
-g_k^T d_k = \sum_{i=1}^{m} \beta_k^{(k)}_{i-1} g_k^T g_{k-i+1}
\]
\[
= \left( 1 - \sum_{i=2}^{m} |\beta_k^{(k)}_{i-1}| \right) \|g_k\|^2 + \sum_{k=2}^{m} \beta_k^{(k)}_{k-i+1} g_k^T g_{k-i+1}
\]
\[
= \|g_k\|^2 - \sum_{i=2}^{m} |\beta_k^{(k)}_{k-i+1}| \|g_k\|^2 - |g_k^T g_{k-i+1}|\]
\[\]
\[ \geq \|g_k\|^2 - \sum_{k=2}^{m} s_k \|g_k\|^2 + \|g_k^T g_{k-1}\| \]
\[ = \|g_k\|^2 - \frac{\rho}{m-1} (m-1) \|g_k\|^2 \]
\[ = (1 - \rho) \|g_k\|^2. \]

This completes the proof. \qed

**Lemma 2.2.** For all \( k \geq m \),
\[ \|d_k\| \leq \max_{1 \leq i \leq m} \|g_{k-i+1}\|. \]

**Proof.** At first, we can show that \( \beta_k^{(k)} > 0 \). In fact,
\[ \beta_k^{(k)} = 1 - \sum_{i=2}^{m} |\beta_{k-i+1}^k| \]
\[ = 1 - \sum_{i=2}^{m} s_k^i \]
\[ = 1 - \frac{\rho}{m-1} \sum_{i=2}^{m} \|g_k\|^2 + |g_k^T g_{k-i+1}| \]
\[ \geq 1 - \rho > 0. \]

For \( k \geq m \), we have
\[ \|d_k\| = \left\| \sum_{i=1}^{m} \beta_{k-i+1}^{(k)} g_{k-i+1} \right\| \]
\[ \leq \sum_{i=1}^{m} |\beta_{k-i+1}^{(k)}| \cdot \|g_{k-i+1}\| \]
\[ = \max_{1 \leq i \leq m} \{\|g_{k-i+1}\|\}. \]

This completes the proof. \qed

In order to prove the global convergence of **Algorithm A**, we assume that

(H1) The objective function \( f(x) \) is continuously differentiable and has a lower bound on \( \mathbb{R}^n \).

(H2) The gradient \( g(x) = \nabla f(x) \) is uniformly continuous on an open convex set \( B \) that contains the level set
\[ L(x_0) = \{x \in \mathbb{R}^n | f(x) \leq f(x_0)\}, \]
in which \( x_0 \) is an initial point.

(H2'). The gradient \( g(x) \) is Lipschitz continuous on the open convex set \( B \) that contains \( L(x_0) \), i.e., there exists an \( L > 0 \) such that
\[ \|g(x) - g(y)\| \leq L \|x - y\|, \quad \forall x, y \in B. \quad (7) \]

It is obvious that \( (H2') \) implies \( (H2) \).

3. Global convergence

**Lemma 3.1.** If \( (H2') \) holds, then
\[ \alpha_k \geq \frac{(1 - \mu_2) g_k^T d_k}{L \|d_k\|^2}. \]
Proof. By using the mean value theorem on the left-hand side of (4), there exists $\theta_k \in [0, 1]$ such that
\[
\alpha_k g(x_k + \theta_k \alpha_k d_k)^T d_k = f_{k+1} - f_k \geq \alpha_k \mu_2 g_k^T d_k.
\]
By combining (H2') and the Cauchy–Schwarz inequality, we have
\[
\alpha_k \geq \frac{-(1 - \mu_2)g_k^T d_k}{L \|d_k\|^2}.
\]
This completes the proof. □

Theorem 3.1. If (H1) and (H2') hold and Algorithm A generates an infinite sequence \{x_k\}, then
\[
\sum_{k=0}^{\infty} \frac{\|g_k\|^4}{\gamma_k} < +\infty,
\]
where
\[
\gamma_k = \max_{1 \leq i \leq m} \|g_{k-i+1}\|^2.
\]
Proof. By the right-hand side inequality of (4) and Lemmas 2.1, 2.2 and 3.1, we have
\[
f_k - f_{k+1} \geq -(\alpha_k \mu_1 \infty \|d_k\| \|g_k\|)
\]
\[
\geq \frac{\mu_1 (1 - \mu_2)}{L} \cdot \left(\frac{-g_k^T d_k}{\|d_k\|}\right)^2 \quad \text{by Lemma 3.1}
\]
\[
\geq \frac{\mu_1 (1 - \mu_2)}{L} \cdot \frac{(g_k^T d_k)^2}{\gamma_k} \quad \text{by Lemma 2.2}
\]
\[
\geq \frac{\mu_1 (1 - \mu_2)}{L} \cdot \frac{(1 - \rho)^2 \|g_k\|^4}{\gamma_k} \quad \text{by Lemma 2.1}
\]
\[
\geq \frac{\mu_1 (1 - \mu_2)(1 - \rho)^2}{L} \cdot \frac{\|g_k\|^4}{\gamma_k}
\]
\[
= \eta \frac{\|g_k\|^4}{\gamma_k},
\]
where
\[
\eta = \frac{\mu_1 (1 - \mu_2)(1 - \rho)^2}{L}.
\]
The above inequality shows that \{f_k\} is a decreasing sequence and (H1) guarantees that \{f_k\} has a bound from below. Therefore, \{f_k\} is a convergent sequence. Thus (8) holds. The proof is completed. □

Lemma 3.2. If the conditions in Theorem 3.1 hold then \{\|g_k\|\} has a bound and thus \gamma_k defined in Theorem 1 has also a bound.
Proof. For the contrary, if \{\|g_k\|\} has no bound then we can deduce a contradiction. Let
\[
\delta_k = \max_{0 \leq i \leq k} \{\|g_i\|^2\}.
\]
Assume that \( \{ \| g_k \| \} \) has no bound then
\[
\lim_{k \to \infty} \delta_k = +\infty. \tag{11}
\]

We can prove that there exists an infinite subset \( K \) of \( \{0, 1, 2, \ldots, \} \) such that
\[
\delta_k = \| g_k \|^2, \quad k \in K. \tag{12}
\]
In fact, if there is no such infinite subset \( K \) of \( \{0, 1, 2, \ldots, \} \) such that (12) holds, then there exists a \( k' \) such that
\[
\delta_k > \| g_k \|^2, \quad k \geq k'.
\]
This implies that
\[
\| g_k \|^2 < \delta_k = \max\{ \| g_k \|^2, \delta_{k-1} \} = \delta_{k-1} = \cdots = \delta_{k'} < +\infty,
\]
which contradicts (11).

By Theorem 3.1, (12) and noting that \( \gamma_k \leq \delta_k \), we have
\[
+\infty > \sum_{k=0}^{\infty} \frac{\| g_k \|^4}{\gamma_k} \\
\quad \geq \sum_{k \in K} \frac{\| g_k \|^4}{\gamma_k} \\
\quad \geq \sum_{k \in K} \frac{\| g_k \|^4}{\delta_k} \\
\quad = \sum_{k \in K} \| g_k \|^2.
\]
Thus
\[
\lim_{k \in K, k \to \infty} \| g_k \|^2 = 0,
\]
which contradicts (12) and (11). This shows that \( \{ \| g_k \| \} \) has a bound and thus \( \{ \gamma_k \} \) also has a bound. The proof is completed. \( \square \)

**Theorem 3.2.** If the conditions in Theorem 3.1 hold then
\[
\lim_{k \to \infty} \| g_k \| = 0. \tag{13}
\]

**Proof.** By Lemma 3.2, there is an \( M > 0 \) such that
\[
\| g_k \| \leq M, \quad \forall k. \tag{14}
\]
Therefore, by noting the definition of \( \gamma_k \) in Theorem 3.1, it follows that
\[
\gamma_k \leq M^2, \quad \forall k. \tag{15}
\]
By Theorem 3.1, (8) and (15), we have
\[
+\infty > \sum_{k=0}^{\infty} \frac{\| g_k \|^4}{\gamma_k} \\
\quad \geq \sum_{k=0}^{\infty} \frac{\| g_k \|^4}{M^2}.
\]
Thus, (13) holds. The proof is completed. \( \square \)
4. Further convergence properties

As we can see, the condition (H2) is weaker than \((H2')\). We now prove the global convergence of Algorithm A under the weaker conditions (H1) and (H2).

**Lemma 4.1.** Assume that (H1) and (H2) hold. Algorithm A generates an infinite sequence \(\{x_k\}\). Then \(\|g_k\|\) has an upper bound and thus \(\{\gamma_k\}\) and \(\{|d_k|\}\) also have an upper bound.

**Proof.** By Lemma 2.1 and the Cauchy–Schwarz inequality we have

\[
|d_k| \geq (1 - \rho)\|g_k\|.
\]

By combining Lemma 2.2 we have

\[
(1 - \rho)^2\|g_k\|^2 \leq |d_k|^2 \leq \gamma_k, \quad \forall k.
\]  \hspace{1cm} (16)

To the contrary, assuming that \(\|g_k\|\) has no bound, then there exists an infinite subset \(K\) such that (11) and (12) hold.

By (H1), (11) and (16), and the right-hand side inequality of (4), we have

\[
+\infty > \sum_{k=0}^{\infty} (f_k - f_{k+1})
\]

\[
\geq -\mu_1 \sum_{k \in K} \alpha_k \|g_k\| d_k
\]

\[
\geq \mu_1 (1 - \rho) \sum_{k \in K} \alpha_k \|g_k\|^2
\]

\[
= \mu_1 (1 - \rho) \sum_{i \in K} \alpha_k \gamma_k
\]

\[
\geq \mu_1 (1 - \rho) \sum_{k \in K} \alpha_k \|d_k\|^2.
\]

Thus

\[
\lim_{k \in K, k \to \infty} \alpha_k \|d_k\|^2 = 0.
\]  \hspace{1cm} (17)

By (11), (12) and (16) we have

\[
\lim_{k \in K, k \to \infty} |d_k| = +\infty.
\]  \hspace{1cm} (18)

By (17) and (18) we have

\[
\lim_{k \in K, k \to \infty} (\alpha_k \|d_k\|) = 0.
\]  \hspace{1cm} (19)

By using the mean value theorem on the left-hand side of (4), there exists a \(\theta_k \in [0, 1]\) such that

\[
\alpha_k g(x_k + \theta_k \alpha_k d_k)^T d_k = f_{k+1} - f_k \geq \alpha_k \mu_2 g_k^T d_k.
\]

Thus

\[
g(x_k + \theta_k \alpha_k d_k)^T d_k \geq \mu_2 g_k^T d_k.
\]  \hspace{1cm} (20)

By (12) and Lemma 2.2 we have

\[
|d_k|^2 \leq \max_{1 \leq i \leq m} \|g_{k-i+1}\|^2 \leq \delta_k = \|g_k\|^2, \quad k \in K.
\]
By combining (20), the Cauchy–Schwarz inequality and Lemma 2.1, we have
\[ \|g(x_k + \theta_k \alpha_k d_k) - g_k\| \cdot \|d_k\| \geq (g(x_k + \theta_k \alpha_k d_k) - g_k)^T d_k \]
\[ \geq -(1 - \mu_2) g_k^T d_k \]
\[ \geq (1 - \mu_2)(1 - \rho)\|g_k\|^2 \]
\[ \geq (1 - \mu_2)(1 - \rho)\|d_k\| \cdot \|g_k\|, \quad k \in K. \]

Thus
\[ \|g(x_k + \theta_k \alpha_k d_k) - g_k\| \geq (1 - \mu_2)(1 - \rho)\|g_k\|, \quad k \in K. \]

By (H2) and (19) we get
\[ \lim_{k \in K, k \to +\infty} \|g_k\| = 0, \]
which contradicts (11) and (12). This shows that \(\{\|g_k\|\}\) has an upper bound. Therefore, \(\{\|d_k\|\}\) also has an upper bound. The proof is completed. □

**Theorem 4.1.** Assume that (H1) and (H2) hold. Algorithm A generates an infinite sequence \(\{x_k\}\). Then
\[ \lim_{k \to \infty} \|g_k\| = 0. \]

**Proof.** To the contrary, assume that there is an infinite subset \(K\) of \(\{0, 1, 2, \ldots\}\) such that
\[ \|g_k\| \geq \epsilon, \quad \forall k \in K, \] (21)
in which \(\epsilon > 0\). By (H2), the right-hand side inequality of (4) and Lemma 2.1, we have
\[ +\infty > \sum_{k=0}^{\infty} (f_k - f_{k+1}) \]
\[ \geq \sum_{k \in K} (f_k - f_{k+1}) \]
\[ \geq -\mu_1 \sum_{k \in K} \alpha_k g_k^T d_k \]
\[ \geq \mu_1 (1 - \rho) \sum_{k \in K} \alpha_k \|g_k\|^2 \]
\[ \geq \mu_1 (1 - \rho) \epsilon^2 \sum_{k \in K} \alpha_k. \]

Thus
\[ \lim_{k \in K, k \to \infty} \alpha_k = 0. \] (22)

By Lemma 4.1 we have
\[ \lim_{k \in K, k \to \infty} (\alpha_k \|d_k\|) = 0. \] (23)

By (20), Lemma 2.1 and the Cauchy–Schwarz inequality, we have
\[ \|g(x_k + \theta_k \alpha_k d_k) - g_k\| \cdot \|d_k\| \geq (g(x_k + \theta_k \alpha_k d_k) - g_k)^T d_k \]
\[ \geq -(1 - \mu_2) g_k^T d_k \]
\[ \geq (1 - \mu_2)(1 - \rho)\|g_k\|^2. \]

By Lemma 4.1, (23), (H2) and the above inequality, we have
\[ \lim_{k \in K, k \to \infty} \|g_k\| = 0, \]
which contradicts (21). The proof is completed. □
5. Linear convergence rate

Assumption. (H3) \( f(x) \) is uniformly convex and twice continuously differentiable.

In fact, Assumption (H3) implies (H1) and (H2).

Lemma 5.1. If (H3) holds, then \( f(x) \) has the following properties

1. \( f(x) \) has a unique minimizer on \( \mathbb{R}^n \), \( x^* \) say.
2. The level set \( L(x_0) = \{ x | f(x) \leq f(x_0) \} \) is bounded.
3. There exist \( m' > 0 \) and \( M' > 0 \) such that
   \[ m'\|x - x^*\|^2 \leq f(x) - f(x^*) \leq \frac{1}{2}M'\|x - x^*\|^2, \]
   \[ m'\|x - x^*\| \leq \|g(x)\| \leq M'\|x - x^*\|. \]
4. Assumptions (H1) and (H2') hold.

Proof. The proof can be seen in [24]. □

Theorem 5.1. If (H3) holds, then \( \{x_k\} \to x^* \), where \( x^* \) is the unique minimizer of \( f \). Further, either there exists an infinite subset \( K \subset \{m, m+1, \ldots\} \) and \( i_0 : 2 \leq i_0 \leq m \) such that

\[ \lim_{k \to \infty, k \in K} \frac{\|g_k\|}{\|g_k - i_0 + 1\|} = 0, \]

or

\[ \limsup_{k \to \infty} \|x_k - x^*\|^{\frac{1}{2}} < 1. \]

Proof. By the proof of Theorem 3.1, we have

\[ f_k - f_{k+1} \geq \eta - \frac{\eta\|g_k\|^4}{\gamma_k} \geq \eta \frac{\|g_k\|^2}{1 + \gamma_k/\|g_k\|^2}. \]

If \( \{\gamma_k/\|g_k\|^2\} \) has no bound, then there exists an infinite subset \( K \) and \( i_0 : 2 \leq i_0 \leq m \) such that

\[ \lim_{k \to \infty, k \in K} \frac{\|g_k - i_0 + 1\|}{\|g_k\|} = \infty, \]

and thus

\[ \lim_{k \to \infty, k \in K} \frac{\|g_k\|}{\|g_k - i_0 + 1\|} = 0. \]

If \( \{\gamma_k/\|g_k\|^2\} \) has a bound, i.e., there exists a \( \mu > 0 \) such that

\[ \frac{\gamma_k}{\|g_k\|^2} \leq \mu, \]

then

\[ f_k - f_{k+1} \geq \eta - \frac{\eta}{1 + \mu} \|g_k\|^2 = \eta_0 \|g_k\|^2, \]

where \( \eta_0 = \eta/(1 + \mu) \) and \( \eta \) is as in the proof of Theorem 3.1. By (H3), Lemma 5.1, and Theorem 3.2, we have \( \{x_k\} \to x^* \) with \( x^* \) being the unique minimizer of \( f \). By (24), the remainder proof follows from [24]. This completes the proof. □
6. Numerical results

The conjugate gradient method takes the form (3) in which
\[
\beta_k^{FR} = \frac{\|g_k\|^2}{\|g_{k-1}\|^2}, \quad \beta_k^{PRP} = \frac{g_k^T (g_k - g_{k-1})}{\|g_{k-1}\|^2}, \quad \beta_k^{HS} = \frac{g_k^T (g_k - g_{k-1})}{d_k^T g_{k-1}},
\]
and its corresponding method is called FR, PRP, HS conjugate gradient method respectively [1,2,4]. Some other formulae can be seen in the literature [25–28]. The new method in the paper is denoted by NM, and the steepest descent method by SM. All these methods have the same property that avoids the overhead and evaluation of the second derivative of \(f\), the storage and computation of the matrix associated with Newton-type methods.

For non-quadratic objective functions, we use Goldstein line search to choose the step size \(\alpha_k\) in steepest descent method, FR, PRP, HS conjugate gradient method, etc. How to implement the Goldstein line search is very important in practical computation. In fact, the Goldstein line search contains two inequalities
\[
\alpha_k \mu_2 g_k^T d_k \leq f(x_k + \alpha_k d_k) - f_k, \quad (25)
\]
and
\[
f(x_k + \alpha_k d_k) - f_k \leq \alpha_k \mu_1 g_k^T d_k, \quad (26)
\]
where \(0 < \mu_1 < \mu_2 < 1\). There are two questions regarding the line search. One is if \(\alpha_k\) exists. The other is how to solve the system of two inequalities. The first question was answered in [21,24] and the Goldstein line search is well-defined. We can solve (25) and (26) in a finite number of steps. The procedure is described as follows.

**Goldstein line search.** Given \(\eta_1, \eta_2 \in (0,1), s > 0\).

Step 1. \(\alpha_k = s\);
Step 2. If (25) and (26) hold then stop else go to Step 3;
Step 3. If (25) holds but (26) does not hold then \(\alpha_k := \eta_2 \alpha_k\), go to Step 2;
Step 4. If (26) holds but (25) does not hold then \(\alpha_k := \eta_2^{-1} \alpha_k\), go to Step 2;
In practical computation, we take \(\eta_1 = 0.75\) and \(\eta_2 = 0.25\).

The numerical comparisons include:

1. The number of iterations for attaining the same precision \(\|g_k\| \leq \text{eps}\) in which \(\text{eps} = 10^{-9}\) is a computational precision;
2. The number of functional evaluations for attaining the same precision.

We choose several test problems for our numerical experiments.

**Test 1.** [29]
\[
f(x) = (x_1 + 10x_2)^4 + 5(x_3 - x_4)^4 + (x_2 - 2x_3)^4 + 10(x_1 - x_4)^4, \quad x_0 = (2, 2, -2, -2)^T, \quad x^* = (0, 0, 0, 0)^T, \quad f^* = 0.
\]

**Test 2.** [30]
\[
f = (1 - x_1)^2 + (1 - x_{10})^2 + \sum_{i=1}^{9} (x_i^2 - x_{i+1})^2, \quad x_0 = (-2, \ldots, -2)^T, \quad x^* = (1, \ldots, 1)^T, \quad f^* = 0.
\]

**Test 3.** Extended Powell function [31]:
\[
f = \sum_{i=1}^{n-3} [(x_i + 10x_{i+1})^2 + 5(x_{i+2} - x_{i+3})^2 + (x_{i+1} - 2x_{i+2})^4 + 10(x_i - x_{i+3})^4], \quad x_0 = (3, -1, 0, 1, \ldots, 3, -1, 0, 1)^T, \quad x^* = (0, 0, \ldots, 0)^T, \quad f^* = 0.
\]
Table 1
The number of iterations and functional evaluations

<table>
<thead>
<tr>
<th>T</th>
<th>NM ($m = 3$)</th>
<th>SM</th>
<th>FR</th>
<th>PRP</th>
<th>HS</th>
</tr>
</thead>
<tbody>
<tr>
<td>T1</td>
<td>16/205</td>
<td>67/2431</td>
<td>14/820</td>
<td>17/743</td>
<td>17/932</td>
</tr>
<tr>
<td>T2</td>
<td>31/566</td>
<td>146/609</td>
<td>589/9718</td>
<td>75/595</td>
<td>168/5967</td>
</tr>
<tr>
<td>T3</td>
<td>42/115</td>
<td>190/1903</td>
<td>58/663</td>
<td>63/934</td>
<td>64/715</td>
</tr>
<tr>
<td>T3'</td>
<td>321/1614</td>
<td>498/3433</td>
<td>538/9862</td>
<td>392/2981</td>
<td>476/7485</td>
</tr>
<tr>
<td>T4</td>
<td>124/232</td>
<td>512/3639</td>
<td>428/2867</td>
<td>367/923</td>
<td>438/981</td>
</tr>
<tr>
<td>T5</td>
<td>112/189</td>
<td>284/2471</td>
<td>518/5211</td>
<td>321/738</td>
<td>383/736</td>
</tr>
</tbody>
</table>

Table 2
The numerical results of NM for different $m$

<table>
<thead>
<tr>
<th>T</th>
<th>NM ($m = 3$)</th>
<th>$m = 2$</th>
<th>$m = 4$</th>
<th>$m = 5$</th>
<th>$m = 6$</th>
</tr>
</thead>
<tbody>
<tr>
<td>T1</td>
<td>16/205</td>
<td>16/86</td>
<td>18/220</td>
<td>18/243</td>
<td>21/234</td>
</tr>
<tr>
<td>T2</td>
<td>31/566</td>
<td>27/413</td>
<td>38/428</td>
<td>45/521</td>
<td>48/574</td>
</tr>
<tr>
<td>T3</td>
<td>42/115</td>
<td>37/93</td>
<td>48/163</td>
<td>52/234</td>
<td>54/265</td>
</tr>
<tr>
<td>T3'</td>
<td>321/1614</td>
<td>295/1439</td>
<td>331/1865</td>
<td>352/1973</td>
<td>380/2218</td>
</tr>
<tr>
<td>T4</td>
<td>124/232</td>
<td>118/187</td>
<td>143/184</td>
<td>156/192</td>
<td>138/185</td>
</tr>
<tr>
<td>T5</td>
<td>112/189</td>
<td>128/171</td>
<td>131/211</td>
<td>147/238</td>
<td>162/236</td>
</tr>
</tbody>
</table>

Test 4. Brown almost linear function (problem (27) in [32])

$$f(x) = \sum_{i=1}^{n-1} \left[ x_i + \sum_{j=1}^{n} -(n + 1) \right] + \left[ \left( \prod_{i=1}^{n} \right) - 1 \right]^2, \quad n = 30,000, \quad [x_0]_i = 0.5.$$

Test 5. Linear function – rank 1 (problem (33) in [32], with modified initial values)

$$f(x) = \sum_{i=1}^{m} \left[ i \left( \sum_{j=1}^{n} jx_j \right) - 1 \right]^2, \quad m \geq n, \quad n = 30,000, \quad m = 30,000, \quad [x_0]_i = 1/i.$$

We take $\rho = 0.85$, $\mu_1 = 0.38$ and $\mu_2 = 0.75$ in the algorithm and $m = 3$. The numerical results are reported in Table 1.

In Tables 1 and 2, in each pair of numbers, the first number denotes the number of iterations, and the second number denotes the number of functional evaluations. We take $n = 1000$ in Test 3. When we take $n = 30,000$ in Test 3, the corresponding problem is denoted by Test $3'$. The computational results show that the new method in the paper is very stable in practical computation.

It can be seen from Table 1 that the new memory gradient method ($m = 3$) is superior to conjugate gradient methods in many situations. Firstly, like FR, PRP, HS, and steepest descent method, the new method in the paper avoids the evaluation of second derivatives of objective functions. Secondly, the storage and computation of matrices associated with Newton-type method is avoided at each iteration. The last but not the least important thing is that the new method needs fewer iterations and fewer evaluations of $f$ than FR, PRP, HS, and steepest descent method, etc., when the iteration process reaches the same precision. We can also find that the new memory gradient method is stable in practical computation.

From Table 2, we can see that the smaller the $m$, the better the performance of the new memory gradient method. However, the larger the $m$, the more stable the new memory gradient method. Therefore, we should choose small $m$ if the minimization problem to be solved is well-conditioned. For ill-conditioned minimization problems, we should choose greater $m$ in practical computation to make the new memory gradient method converge stably.
7. Conclusions

In this paper, we proposed a multi-step memory gradient method with Goldstein line search for unconstrained optimization problems and proved its global convergence under mild conditions. We also proved the linear convergence rate of the new method when the objective function is uniformly convex. Numerical results showed that the new algorithm is suitable to solve large-scale optimization problems and more stable than other similar methods in practical computation.

For the future research, we should combine the super-memory gradient method and multi-step quasi-Newton methods [5–9,33] and establish some stable and efficient memory gradient methods for unconstrained optimization problems. Moreover, we can use Barzilai–Borwein approach [34–38] to choose an available step size for the super-memory gradient method at each iteration.

Acknowledgements

The authors would like to thank two anonymous referees and the editor for their careful reading and the valuable suggestions that greatly improved the paper. Special thanks go to one referee who pointed out the proof error of Lemma 4 and simplified the proof of Theorem 2.

References

[29] H.Y. Huang, J.P. Chambliss, Quadratically convergent algorithms and one-dimensional search schemes, J. Optim. Theory Appl. 11 (2) (1973) 175–188.