A SCALED CONJUGATE GRADIENT METHOD WITH MOVING ASYMPOTOTES FOR UNCONSTRAINED OPTIMIZATION PROBLEMS

GUANGHUI ZHOU
Nanjing University of Aeronautics and Astronautics
Nanjing, Jiangsu Province, 211106, China
and
Huaibei Normal University
Huaibei, Anhui Province, 235000, China

QIN NI
Nanjing University of Aeronautics and Astronautics
Nanjing, Jiangsu Province, 211106, China

MEILAN ZENG
Nanjing University of Aeronautics and Astronautics
Nanjing, Jiangsu Province, 211106, China
and
Hubei Engineering University
Xiaogan, Hubei Province, 432000, China

(Communicated by Masao Fukushima)

Abstract. In this paper, a scaled method that combines the conjugate gradient with moving asymptotes is presented for solving the large-scaled nonlinear unconstrained optimization problem. A diagonal matrix is obtained by the moving asymptote technique, and a scaled gradient is determined by multiplying the gradient with the diagonal matrix. The search direction is either a scaled conjugate gradient direction or a negative scaled gradient direction under different conditions. This direction is sufficient descent if the step size satisfies the strong Wolfe condition. A global convergence analysis of this method is also provided. The numerical results show that the scaled method is efficient for solving some large-scaled nonlinear problems.

1. Introduction. We consider the unconstrained optimization problem

$$\min_{x \in \mathbb{R}^n} f(x),$$

(1)

where $f : \mathbb{R}^n \to \mathbb{R}$ is continuously differentiable. The gradient of $f(x)$ is denoted by $g(x) \equiv \nabla f(x)$.

2010 Mathematics Subject Classification. Primary: 65K05; Secondary: 90C30, 49M37.

Key words and phrases. Scaling, conjugate gradient, moving asymptotes, the Wolfe condition, unconstrained optimization.

The authors are supported by the Natural Science Foundation of China (11471159,11571169), the Natural Science Foundation of Jiangsu Province (BK20141409), the Education Department Foundation of Anhui Province(KJ2016A651,2014jyxm161), and the Science and Technology Foundation of the Department of Education of Hubei Province (D20152701).
There are many efficient methods for solving problem (1), such as the conjugate gradient (CG) method, the BFGS method, the moving asymptotes (MA) method, etc. In this paper, we demonstrate a new method which combines the CG with moving asymptotes (MA) for solving large-scaled nonlinear unconstrained optimization problems.

As is well known, the CG method is one of the classical methods for unconstrained optimization. It has been proved effective for solving large-scale unconstrained optimization problems because it needs no storage of matrices. In the conjugate gradient method, the iterates \( \{ x_k \} \) have the following format

\[
x_{k+1} = x_k + \alpha_k d_k,
\]

where the positive scalar \( \alpha_k \) is called the step size, the vector \( d_k \) is called the search direction that is defined by

\[
d_k = \begin{cases} -g_0, & \text{if } k = 0, \\ -g_k + \beta_k d_{k-1}, & \text{if } k \geq 1 \\ \end{cases}
\]

with \( g_k \equiv g(x_k) \) and a scalar parameter \( \beta_k \) characterizing the conjugate gradient method. The best-known expressions of \( \beta_k \) are Hestenes–Stiefel (HS) [10], Fletcher–Reeves (FR) [6], Polak–Ribiere–Polyak (PRP) [15, 16] and Dai–Yuan (DY) [3] formulas, which are defined by

\[
\beta_{HS}^k = \frac{g_k^T y_{k-1}}{d_{k-1}^T y_{k-1}}, \quad \beta_{FR}^k = \frac{\| g_k \|^2}{\| g_{k-1} \|^2}, \quad \beta_{PRP}^k = \frac{g_k^T y_{k-1}}{\| g_{k-1} \|^2}, \quad \beta_{DY}^k = \frac{\| g_k \|^2}{d_{k-1}^T y_{k-1}},
\]

respectively, where \( y_k = g_{k+1} - g_k \) and \( \| \cdot \| \) denotes the Euclidean norm.

A popular inexact line search is the Wolfe condition, in which the step size \( \alpha_k \) satisfies

\[
f(x_k) - f(x_k + \alpha_k d_k) \geq -\delta \alpha_k g_k^T d_k, \quad (5)
\]

\[
g(x_k + \alpha_k d_k)^T d_k \geq \sigma g_k^T d_k \quad (6)
\]

with \( 0 < \delta < \sigma < 1 \). Many numerical methods are proved to be convergent under the strong Wolfe condition, that is, the step size \( \alpha_k \) satisfies (5) and (6) and

\[
\| g(x_k + \alpha_k d_k) \| d_k \leq -\sigma g_k^T d_k. \quad (7)
\]

In recent years, the CG method has attracted new attentions. Some scholars have further investigated the method in more regions. For example, Narushima et al. [13] proposed a three-term conjugate gradient method in which the search direction satisfies the sufficient descent condition, Al-Baali et al. [1] extended this method to a two-parameter family of three-term conjugate gradient methods which can be used to control the magnitude of the directional derivative, Dai and Kou [4] sought the conjugate gradient direction closest to the direction of the scaled memoryless BFGS method and proposed a family of conjugate gradient methods for unconstrained optimization, Hager and Zhang [9] developed a limited memory conjugate gradient method to find out the loss of orthogonality that can occur in ill-conditioned optimization problems and then correct it, Nakamura, Narushima and Yabe [12] considered a unified formula of parameters which establishes the sufficient descent condition, Zhou and Zhou [19] proved the strongly global convergence property of a modified three-term HS conjugate gradient method for nonconvex optimization by the use of the backtracking type line search, and etc.

The MA method was first presented by Svanberg in [17] and later frequently used for structural optimization. Ni and Wang studied the unconstrained optimization
problem (1) using the MA with a trust region technique in [14] and [18], respectively. They designed a new subproblem and some new rules to control the parameters of moving asymptotes. The numerical results showed that their method is capable of processing large-scale optimization problems.

In this paper, we propose a new method for solving large-scale nonlinear unconstrained optimization problems. We shall establish a new search direction with the help of scaling thought introduced by Luenberger and Ye in [11]. More precisely, we firstly obtain a diagonal matrix, called a scaled matrix, in the moving asymptotes method. Then we apply this matrix to our new search direction, which is called a scaled conjugate gradient direction. In our method, the step size satisfies the Wolfe condition or the strong Wolfe condition. This method can be viewed as a combination of the CG and the MA. In fact, they are two particular cases of our method. Our purpose is to obtain a new method which may have the advantages of both the CG and the MA.

2. The algorithm. Wang and Ni [18] studied unconstrained optimization by using the MA method. They obtained a descent direction by solving a convex separable subproblem of moving asymptotes in each iteration. The subproblem is

\[
\min_{x \in \mathbb{R}^n} \ m(x, \delta) = f(x) + \sum_{i=1}^{n} \phi_i(\delta_i) \quad \text{s.t.} \quad |\delta_i| \leq \Delta_i, \quad i = 1, 2, \ldots, n, \tag{8}
\]

where \( \delta_i \) is the \( i \)-th element of the direction \( \delta \), \( \Delta_i (i = 1, 2, \ldots, n) \) are trust region radii and

\[
\phi_i(\delta_i) = \begin{cases} 
\frac{\| \tilde{g}_i \| \Delta_i}{a_i} + \frac{\varepsilon_i \delta_i^2}{a_i^2 - \delta_i^2} & \text{if } i \in I_+,
\frac{\| \tilde{g}_i \| \Delta_i}{a_i} + \frac{\varepsilon_i \delta_i^2}{a_i^2 - \delta_i^2} & \text{if } i \in I_-
\end{cases}
\]

where \( \tilde{g}_i \) is the \( i \)-th element of the gradient \( \tilde{g} \), \( I_+ = \{ i \mid \tilde{g}_i > 0 \}, I_- = \{ i \mid \tilde{g}_i < 0 \} \), \( a_i = \Delta_i + \eta_i, \eta_i > 0 \) and \( \varepsilon_i > 0 \). Wang and Ni replaced general moving asymptotes with the trust region radius \( \Delta_i \) and parameter \( \eta_i \), so they showed a reasonable way to control the parameters of moving asymptotes.

Wang and Ni [18] pointed out that the function \( m(x, \delta) \) in (8) is a first-order approximation of \( f(x) \) and the subproblem (8) is equivalent to \( n \) independent one-dimensional bound constrained subproblems

\[
\begin{align*}
\min_{\delta_i} & \quad \phi_i(\delta_i) \\
\text{s.t.} & \quad |\delta_i| \leq \Delta_i,
\end{align*}
\]

where \( i = 1, 2, \ldots, n, \Delta_1, \ldots, \Delta_n \) are trust region radii. The stationary point of \( \phi_i(\delta_i) \) is given by \( \delta_i = -\gamma_i \tilde{g}_i \), where

\[
\gamma_i = \frac{2\tilde{g}_i^2}{(\varepsilon_i + 2a_i |\tilde{g}_i|) (1 + \lambda_i)^2}, \quad \lambda_i = \sqrt{\frac{\varepsilon_i}{\varepsilon_i + 2a_i |\tilde{g}_i|}}, \quad a_i = \Delta_i + \eta_i
\]

with positive parameters \( \varepsilon_i \) and \( \eta_i \). The solution of (8) is shown to be

\[
\delta_i = -\gamma_i \tilde{g}_i, \quad \gamma_i = \begin{cases} 
\frac{\gamma_i}{\tilde{g}_i^2} & \text{if } i \in J_1,
\frac{\lambda_i}{\tilde{g}_i} & \text{if } i \in J_2,
\frac{\Delta_i}{-\tilde{g}_i} & \text{if } i \in J_3.
\end{cases} \tag{9}
\]

where \( J_1 = \{ i \mid \delta_i \in [-\Delta_i, \Delta_i] \} \), \( J_2 = \{ i \mid \delta_i < -\Delta_i \} \), \( J_3 = \{ i \mid \delta_i > \Delta_i \} \).
The expression [9] implies $\gamma_i > 0$, $i = 1, 2, \cdots, n$. Furthermore, we define a diagonal matrix $\Lambda$ by

$$\Lambda = \text{diag}(\gamma_1, \gamma_2, \cdots, \gamma_n),$$  \hfill (10)

where $\gamma_i(i = 1, 2, \cdots, n)$ is determined by [9]. Then $\Lambda$ is positive definite.

We can say that the MA method in [18] is essentially a modified steepest descent method since the solution of (8) can be expressed as $\tilde{d} = -\Lambda g$. We note that this direction can also be obtained by the linear transformation $x = \Lambda^2 z$ and the negative gradient of $\phi(z) \equiv f(\Lambda^2 z)$.

Now let us expatiate and use an idea, which has been elaborated in [11] by Luenberger and Ye, to derive our new conjugate gradient method. Suppose that $x = Az$ is a linear invertible transformation, where $A$ is an invertible $n \times n$ matrix. Then we have

$$\alpha_k d_k = x_{k+1} - x_k = A(z_{k+1} - z_k) = \tilde{a}_k A \tilde{d}_k.$$  

To simplify the deduction, we assume $\alpha_k = \tilde{a}_k$. Let $\tilde{g}$ be the gradient of the function $f(Az)$ with respect to the variable $z$, i.e., $\tilde{g}(z) : = \nabla_z f(Az)$, then $\tilde{g}(z) = ATg(x)$.

The search direction of the conjugate gradient with respect to the variable $z$ can be expressed as

$$\tilde{d}_k = -\tilde{g}_k + \tilde{\beta}_k \tilde{d}_{k-1}.$$  

We choose one formula of (4) as the expression of $\tilde{\beta}_k$, for example,

$$\tilde{\beta}_{k}^{\text{DY}} = \frac{\tilde{g}^T_k \tilde{g}_k}{d_{k-1}^T (\tilde{g}_k - \tilde{g}_{k-1})} = \frac{\tilde{g}_k^T A A^T g_k}{d_{k-1}^T y_{k-1}}.$$  

From above three equations,

$$d_k = -AA^T g_k + \frac{\tilde{g}_k^T A A^T g_k}{d_{k-1}^T y_{k-1}} d_{k-1}.$$  

If $A$ is chosen as $A \frac{2}{z}$, then

$$d_k = -Ag_k + \frac{\tilde{g}_k^T A g_k}{d_{k-1}^T y_{k-1}} d_{k-1}.$$  

Furthermore, if we allow $\Lambda$ to vary in each iteration, then the search direction can be expressed as

$$d_k = \begin{cases} 
-g_0 & \text{if } k = 0, \\
-\Lambda_k g_k + \frac{g_k^T \Lambda_k g_k}{d_{k-1}^T y_{k-1}} d_{k-1} & \text{if } k \geq 1, 
\end{cases}$$  \hfill (11)

where $\Lambda_k$ is defined by [10].

In order to control the change of diagonal elements $\gamma_i(i = 1, 2, \cdots, n)$ of the matrix $\Lambda_k$ within a particular range, we will choose two constants $m$ and $M$ satisfying $0 < m < 1 < M$, such that

$$m \leq \gamma_i \leq M$$  \hfill (12)

for $i = 1, 2, \cdots, n$. This restriction can ensure the descent property of the direction (11) in some sense. In order to guarantee the global convergence, we also need to modify the parameters and define the following notations

$$\hat{d}_k = -\Lambda_k g_k + \tilde{\beta}_k d_{k-1}, \quad \tilde{\beta}_k = \frac{g_k^T \Lambda_k g_k}{d_{k-1}^T y_{k-1}}, \quad \theta_k = \frac{g_k^T \Lambda_k g_k}{\|g_k\|^2},$$  \hfill (13)
Hence, our search direction can be expressed as
\[
d_k = \begin{cases} 
-g_0 & \text{if } k = 0, \\
\hat{d}_k & \text{if } k \geq 1, g_k^T \Lambda_k \hat{d}_k \geq \theta_k g_k^T \hat{d}_k, \\
-\Lambda_k g_k, & \text{otherwise,}
\end{cases}
\] (14)
where \( \hat{d}_k \) and \( \theta_k \) are defined by (13).

Now we can give a scaled method combining the conjugate gradient with moving asymptotes (denoted as the SMCGMA method for convenience).

**Algorithm SMCGMA**

**Step 1** *Initialization.* Choose \( x_0 \in \mathbb{R}^n \), and set \( k = 0 \). Take \( \epsilon > 0 \), \( K > 1 \), \( \Delta_i^0 > 0 \), \( i = 1, \ldots, n \), \( 0 \leq c_1 < c_2 < 1 \), \( 0 < m < 1 < M \), \( 0 < \mu_1 \leq \mu_2 < 1 < \mu_3 \).

**Step 2** *Check the convergence condition.* Compute \( g_k \); if \( \|g_k\| \leq \epsilon \), then stop.

**Step 3** *Form the search direction.* Compute \( \eta_i^k \) and \( \varepsilon_i^k \), where
\[
\eta_i^k = \frac{K}{\max \{ |g_i^k|, 0.1 \}}, \quad \varepsilon_i^k = \min \left( 0.1 \eta_i^k, 0.5 \right),
\]
and \( g_i^k \) is the \( i \)-th component of \( g_k \) for \( i = 1, \ldots, n \). Update the scaled matrix \( \Lambda_k \) by (9), (10), and (12). Then compute the search direction \( d_k \) by (14).

**Step 4** *Line search.* Find \( \alpha_k \) which satisfies the Wolfe condition ((5) and (6)) or the strong Wolfe condition ((3) and (7)). Set \( s_k = \alpha_k d_k \).

**Step 5** *Update the trust region radius.* Compute
\[
\Delta_{k+1}^i = \begin{cases} 
[\mu_1 \Delta_i^k, \mu_2 \Delta_i^k] & \text{if } \xi_k < c_1, \\
\Delta_i^k & \text{if } c_1 \leq \xi_k < c_2, \\
(\mu_3 \Delta_i^k, \mu_3 \Delta_i^k] & \text{otherwise,}
\end{cases}
\]
where
\[
\xi_k = \frac{f(x^k + s^k) - f(x^k)}{-\sum_{i=1}^n \phi_i^k(s^k)}.
\]

**Step 6** *Compute the new point.* Set \( x_{k+1} = x_k + s_k \), \( k = k + 1 \), and go to Step 2.

### 3. Convergence analysis
As mentioned above, for unconstrained optimization, Algorithm SMCGMA is a new method which can be considered as a generalization of the Dai-Yuan CG method [3] and the Wang-Ni method [18].

It follows from (14) that
\[
g_k^T \hat{d}_k = -g_k^T \Lambda_k g_k + \frac{g_k^T \Lambda_k g_k}{d_{k-1}^T y_{k-1}} g_k^T d_{k-1}
\]
\[
= g_k^T \Lambda_k g_k \left[ \frac{1}{d_{k-1}^T y_{k-1}} \left( -d_{k-1}^T y_{k-1} + g_k^T d_{k-1} \right) \right]
\]
\[
= g_k^T \Lambda_k g_k \frac{g_k^T d_{k-1}}{d_{k-1}^T y_{k-1}} d_{k-1}
\]
\[
= \hat{\beta} g_k^T d_{k-1},
\]
which implies that
\[
g_k^T \hat{d}_k = \frac{g_k^T \Lambda_k g_k}{g_k^T d_{k-1}} - 1 = \frac{g_k^T \Lambda_k g_k}{s_{k-1} - 1},
\] (16)
where
\[ l_{k-1} = \frac{g_k^T d_{k-1}}{d_{k-1}^T d_{k-1}}. \]
The expression (16) will be helpful for our convergence analysis.

The following lemma shows that the search direction (14) is descent one under the Wolfe condition.

**Lemma 3.1.** Suppose that the sequences \( \{x_k\}, \{g_k\} \) and \( \{d_k\} \) are generated by Algorithm SMCGMA, in which \( \alpha_k \) is obtained by the Wolfe conditions (5) and (6). If \( g_k \neq 0 \) for all \( k \geq 0 \), then
\[ g_k^T d_k < 0. \]  
Proof. (i) If \( k = 0 \), then \( d_0 = -g_0 \), and
\[ g_0^T d_0 = -\|g_0\|^2 < 0. \]

(ii) If \( d_k = -\Lambda_k g_k \), then it follows from the definite positivity of the matrix \( \Lambda_k \) that
\[ g_k^T d_k = -g_k^T \Lambda_k g_k < 0. \]

(iii) If \( d_k = \hat{d}_k \), then we can use induction to show that the relation (17) is also correct in this case. From (i) and (ii), we can make the induction hypothesis that \( g_i^T d_i < 0 \) for \( i = k - 1 \). It follows from the Wolfe line search condition (6) that
\[ g_k^T d_{k-1} - g_{k-1}^T d_{k-1} > 0. \]
By (15),
\[ g_k^T d_k = \frac{g_k^T \Lambda_k g_k}{g_k^T d_{k-1} - g_{k-1}^T d_{k-1}} < 0. \]
The proof is completed.

Furthermore, the direction is sufficient descent if the step size \( \alpha_k \) satisfies the strong Wolfe conditions (5) and (7).

**Lemma 3.2.** Suppose that the sequences \( \{x_k\}, \{g_k\} \) and \( \{d_k\} \) are generated by Algorithm SMCGMA, in which \( \alpha_k \) satisfies the strong Wolfe conditions (5) and (7). If \( g_k \neq 0 \) for all \( k \geq 0 \), then
\[ g_k^T d_k < -\frac{m}{2} \|g_k\|^2, \]  
where \( m \in (0, 1) \) is defined by Step 1 of Algorithm SMCGMA.

Proof. (i) If \( k = 0 \), then \( d_0 = -g_0 \), and
\[ g_0^T d_0 = -\|g_0\|^2 < -\frac{m}{2} \|g_0\|^2. \]

(ii) If \( d_k = \hat{d}_k \), then
\[ g_k^T d_k = g_k^T \hat{d}_k = \frac{g_k^T \Lambda_k g_k}{g_k^T d_{k-1} - g_{k-1}^T d_{k-1}} < \frac{g_k^T \Lambda_k g_k}{g_{k-1}^T d_{k-1}}. \]
It follows from the strong Wolfe condition (7) that
\[ -(\sigma + 1) \leq \frac{g_k^T d_{k-1}}{g_{k-1}^T d_{k-1}} - 1 \leq \sigma - 1 < 0, \]
which implies
\[ g_k^T d_k \leq -\frac{g_k^T \Lambda_k g_k}{(\sigma + 1)} \leq -\frac{m}{2} \|g_k\|^2. \]
(iii) If \( d_k = -\Lambda_k g_k \), then we have
\[
g_k^T d_k = -g_k^T \Lambda_k g_k \leq -m\|g_k\|^2 < -\frac{m}{2}\|g_k\|^2.
\]
The proof is completed. 

**Lemma 3.3.** Suppose that the sequences \( \{x_k\} \), \( \{g_k\} \), \( \{\beta_k\} \) and \( \{d_k\} \) are generated by Algorithm SMCGMA, in which \( \alpha_k \) satisfies the standard Wolfe conditions (5) and (6). If \( g_k \neq 0 \) for all \( k \geq 0 \), then there exists a constant \( c > 1 \) such that
\[
\frac{\|d_k\|^2}{(g_k^T d_k)^2} \leq \frac{\|d_{k-1}\|^2}{(g_{k-1}^T d_{k-1})^2} + \frac{c}{\|g_k\|^2}.
\]

**Proof.** For all \( k \geq 1 \), it follows from (14) that
\[
d_k = -\Lambda_k g_k + \beta_k d_{k-1},
\]
where \( \beta_k \) can be denoted by
\[
\beta_k = \begin{cases} \hat{\beta}_k & \text{if } d_k = \hat{d}_k, \\ 0 & \text{otherwise}. \end{cases}
\]
Taking the Euclidean norm and squaring both sides of (20), we have that
\[
\|d_k\|^2 = \beta^2_k \|d_{k-1}\|^2 - 2d_k^T \Lambda_k g_k - \|\Lambda_k g_k\|^2.
\]
The formulas (17) and (18) imply that \( g_k^T d_k \neq 0 \) for all \( k \). Therefore,
\[
\frac{\|d_k\|^2}{(g_k^T d_k)^2} = \beta^2_k \frac{\|d_{k-1}\|^2}{(g_{k-1}^T d_{k-1})^2} - \frac{2d_k^T \Lambda_k g_k}{g_k^T d_k} \frac{\|\Lambda_k g_k\|^2}{(g_k^T d_k)^2}.
\]

(i) If \( d_k = \hat{d}_k \), then \( g_k^T \Lambda_k \hat{d}_k \geq 0 \), \( g_k^T d_k \) from (14), namely, \( d_k^T \Lambda_k g_k \geq \frac{g_k^T \Lambda_k g_k}{\|g_k\|^2} g_k^T d_k \).

In this case, \( \beta_k = \hat{\beta}_k = \frac{g_k^T \Lambda_k g_k}{\|g_k\|^2} \). Furthermore, from (16),
\[
-2d_k^T \Lambda_k g_k \leq -2g_k^T \Lambda_k g_k \frac{g_k^T d_k}{\|g_k\|^2} \|g_k\|^2 = -2(l_k - 1) \|g_k\|^2.
\]

Since the third term in the right-hand-side of (21) satisfies
\[
-\frac{\|\Lambda_k g_k\|^2}{(g_k^T d_k)^2} = -\frac{\|\Lambda_k g_k\|^2(l_k - 1)^2}{(g_k^T \Lambda_k g_k)^2} \leq -\frac{\|\Lambda_k g_k\|^2(l_k - 1)^2}{\|g_k\|^2} = -\frac{(l_k - 1)^2}{\|g_k\|^2},
\]
from (15) and (21)–(23), we have
\[
\frac{\|d_k\|^2}{(g_k^T d_k)^2} \leq \frac{\|d_{k-1}\|^2}{(g_{k-1}^T d_{k-1})^2} - \frac{2(l_k - 1) - (l_k - 1)^2}{\|g_k\|^2} \leq \frac{1 - l^2}{\|g_k\|^2} + \frac{1}{\|g_k\|^2}.
\]

(ii) If \( d_k = -\Lambda_k g_k \), then from the Algorithm,
\[
\frac{\|d_k\|^2}{(g_k^T d_k)^2} = \frac{\|\Lambda_k g_k\|^2}{g_k^T \Lambda_k g_k} \leq \frac{(\rho\lambda_k)^2\|g_k\|^2}{(\lambda_{\min})^2\|g_k\|^2} \leq M^2\|g_k\|^2.
\]
where \( \rho_{\Lambda_k} \) and \( \lambda_{\min} \) denote the spectral radius and minimum eigenvalue of \( \Lambda_k \), respectively. Let \( c = \frac{M^2}{m^2} \). Then from Step 1 we know that \( c > 1 \), and
\[
\frac{\|d_k\|^2}{(g_k^T d_k)^2} \leq \frac{c}{\|g_k\|^2} \leq \frac{\|d_{k-1}\|^2}{(g_{k-1}^T d_{k-1})^2} + \frac{c}{\|g_k\|^2}.
\]
Hence, in all cases, (19) is correct.

Let us assume that the object function \( f \) satisfies the following basic assumptions.

**Assumptions.**

A1 The function \( f \) is bounded below and is continuously differentiable in a neighbourhood \( \mathcal{N} \) of the level set \( \mathcal{L} := \{ x \in |f(x) \leq f(x_0) \} \), where \( x_0 \) is the starting point;

A2 The gradient \( g(x) \) of \( f \) is Lipschitz continuous in \( \mathcal{N} \), i.e., there exists a constant \( L > 0 \) such that
\[
\|g(x) - g(y)\| \leq L \|x - y\|
\]
for all \( x, y \in \mathcal{N} \).

The following result has been essentially proved by Zoutendijk and Wolfe [3].

**Lemma 3.4.** Suppose that A1 and A2 hold. Assume that \( d_k \) is a descent direction and \( \alpha_k \) is obtained by the Wolfe conditions (5) and (6). Then
\[
\sum_{k \geq 0} (g_k^T d_k)^2 \|d_k\|^2 < +\infty.
\]
We customarily call (24) the Zoutendijk condition.

**Theorem 3.5.** Suppose that A1 and A2 hold, and the sequences \( \{x_k\} \), \( \{g_k\} \) and \( \{d_k\} \) are generated by Algorithm SMCGMA. Then either \( g_k = 0 \) for some \( k \) or
\[
\liminf_{k \to \infty} \|g_k\| = 0.
\]

**Proof.** Suppose that \( g_k \neq 0 \) for all \( k \) and (25) is not true. Then there exists a constant \( c_3 > 0 \) such that
\[
\|g_k\| \geq c_3
\]
for all \( k \) of the iteration.

It follows from (17) and (19) that there exists a constant \( c > 1 \) such that
\[
\sum_{i=1}^{k} \frac{\|d_i\|^2}{(g_i^T d_i)^2} \leq \sum_{i=1}^{k} \frac{\|d_{i-1}\|^2}{(g_{i-1}^T d_{i-1})^2} + \sum_{i=1}^{k} \frac{c}{\|g_i\|^2}.
\]
So, we obtain from (26) and (27) that
\[
\frac{\|d_k\|^2}{(g_k^T d_k)^2} \leq \sum_{i=0}^{k} \frac{c}{\|g_i\|^2} \leq \frac{ck}{c_3}.
\]
This relation is equivalent to
\[
\frac{(g_k^T d_k)^2}{\|d_k\|^2} \geq \frac{c_4}{k},
\]
where \( c_4 = \frac{c_3}{c} \). Summing up these equalities for \( k = 1, 2, \cdots \), we obtain
\[
\sum_{k \geq 0} \frac{(g_k^T d_k)^2}{\|d_k\|^2} = +\infty,
\]
which contradicts the Zoutendijk condition \(24\). Hence, our original assertion \(26\) must be false, giving that either \(g_k = 0\) for some \(k\) or \(25\) holds.

It’s worth mentioning that the sufficient descent condition is not necessary in our convergence analysis.

4. Numerical results. In this section, we show a preliminary computational performance of the SMCGMA implemented in Matlab for solving unconstrained optimization problems.

The experiments are run on a personal computer with a 64 bits processor, 2.5GHz of CPU and 4GB of RAM memory. All the codes are written in the Matlab language and are compiled with this software.

All the test functions are the unconstrained problems coming from the CUTEr library [7] and [2], they are listed in Table 1.

| No. | Function Name                        | No.   | Function Name                        |
|-----|--------------------------------------|-------|--------------------------------------|
| 1   | ARWHEAD                              | 17    | Ext quadratic penalty QP2            |
| 2   | COSINE                               | 18    | Ext quadratic exponential EP1        |
| 3   | EDENSCH                              | 19    | Ext Tridiagonal 2                    |
| 4   | EG2                                  | 20    | Ext DENSCHNF                         |
| 5   | ENGVAL1                              | 21    | HIMMELBG                             |
| 6   | GENROSE                              | 22    | HIMMELH                              |
| 7   | Ext Freudenstein & Roth              | 23    | TOINTGSS                             |
| 8   | Raydan 2                             | 24    | Extended DENSCHNB                    |
| 9   | Ext Tridiagonal 1                    | 25    | LIARWHD                              |
| 10  | Ext TET                              | 26    | Extended Trigonometric               |
| 11  | Diagonal 5                           | 27    | Extended Penalty                     |
| 12  | Diagonal 2                           | 28    | Extended BD1                         |
| 13  | Ext Maratos                          | 29    | Perturbed Quadratic                  |
| 14  | Ext Cliff                            | 30    | Raydan 1                             |
| 15  | Perturbed quadratic diagonal         | 31    | Diagonal 4                           |
| 16  | Ext quadratic penalty QP1            | 32    | QUARTC                               |

We would like to compare our algorithm SMCGMA with the CG-Descent and the DYCG. The other two algorithms are described in the following:

- **CG-Descent**: A conjugate gradient algorithm with guaranteed descent proposed by Hager and Zhang in [5]. It has been proven excellent in recent years.
- **DYCG**: A conjugate gradient algorithm with a strong global convergence property proposed by Dai and Yuan in [3]. Our Algorithm SMCGMA is obtained by scaling the DYCG and the search direction of the SMCGMA, and is closely related to the direction of DYCG.

To make the comparison as fair as possible, all the step sizes in the three algorithms satisfy the Wolfe conditions \(5\) and \(6\). To terminate the executions, we use the criterion \(\|g\| \leq 10^{-3}\) and impose the restriction on the number of iterations less than 500 in all these algorithms. The parameters in Algorithm SMCGMA are chosen as follows:

\[
c_1 = 0.25, c_2 = 0.75, \mu_1 = 0.6, \mu_2 = 0.9,
\]
\[ \mu_3 = 4, K = 5, m = 10^{-1}, M = 10^2. \]

In order to observe the performance of the SMCGMA, the CG-Descent and the DYCG, we choose three different dimensions of each test function. The dimensions are \( n = 10^2 \), \( n = 10^3 \), and \( n = 10^4 \), respectively.

We use the performance profile proposed by Dolan and Moré [5] to compare the efficiency of these algorithms. The performance profiles are displayed in the log_2 scale. According to the numerical results obtained in every dimension, we plot two figures based on the CPU time and iterations, respectively.

**Figure 1.** CPU time performance profiles in a log_2 scale \((n = 10^2)\)

**Figure 2.** Iterations performance profiles in a log_2 scale \((n = 10^2)\)
We can find from Figures 1 and 2 that the SMCGMA, the DYCG and the CG-Descent have similar performances. There are also no large discrepancies on the CPU time when we select the dimension $n = 10^3$ from Figure 3 because every curve crosses the other two, but from Figure 4 it is clear that the SMCGMA needs fewer iterations than the DYCG and the CG-Descent in solving most of the problems.

If we extend the test dimension $n$ to $10^4$, then the SMCGMA stands out from Figures 5 and 6. This can conclude from the fact that the curves of the SMCGMA are above those of the other two methods. It shows that the SMCGMA has the highest probability of being the optimal solver.
An observation that emerges from these figures is that the SMCGMA may be competitive for large-scaled problems.

5. Conclusions. We have proposed the SMCGMA for solving nonlinear unconstrained optimization problems in this paper. It is a new method that combines the conjugate gradient method with the moving asymptotes method. Numerical results show that the SMCGMA is comparable to the DYCQ and the CG-Descent, and the SMCGMA may be more suitable for large-scaled unconstrained optimization problems.
The SMCGMA is obtained by scaling the DYCG. From the numerical results we can say that the Dai-Yuan conjugate gradient method can be improved by the scaling technique. Some competitive algorithms may be obtained if we scale other conjugate gradient methods in the same way. Hence, this scaling technique can be further studied.

Acknowledgments. We are grateful to two anonymous referees for the detailed comments on revising the original draft of the paper. We are also grateful to Professor Jiu Ding for polishing this paper.

REFERENCES

[1] M. Al-Baali, Y. Narushima and H. Yabe, A family of three-term conjugate gradient methods with sufficient descent property for unconstrained optimization, *Computational Optimization and Applications*, 60 (2015), 89–110.

[2] N. Andrei, An unconstrained optimization test functions collection, *Advanced Modeling and Optimization*, 10 (2008), 147–161.

[3] 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] Y. Dai and C. Kou, A nonlinear conjugate gradient algorithm with an optimal property and an improved Wolfe line search, *SIAM Journal on Optimization*, 23 (2013), 296–320.

[5] E. D. Dolan and J. J. Moré, Benchmarking optimization software with performance profiles, *Mathematical Programming*, 91 (2002), 201–213.

[6] R. Fletcher and C. M. Reeves, Function minimization by conjugate gradients, *The Computer Journal*, 7 (1964), 149–154.

[7] N. I. M. Gould, D. Orban and P. L. Toint, CUTEr and SifDec: A constrained and unconstrained testing environment, revisited, *ACM Transactions on Mathematical Software*, 29 (2003), 353–372.

[8] W. Hager and H. Zhang, A new conjugate gradient method with guaranteed descent and an efficient line search, *SIAM Journal on Optimization*, 16 (2005), 170–192.

[9] W. Hager and H. Zhang, The limited memory conjugate gradient method, *SIAM Journal on Optimization*, 23 (2013), 2150–2168.

[10] M. R. Hestenes and E. Stiefel, Methods of conjugate gradients for solving linear systems, *Journal of Research of the National Bureau of Standards*, 49 (1952), 409–436.

[11] D. Luenberger and Y. Ye, *Linear and Nonlinear Programming*, 3rd edition, Springer-Verlag, New York, 2008.

[12] W. Nakamura, Y. Narushima and H. Yabe, Nonlinear conjugate gradient methods with sufficient descent properties for unconstrained optimization, *Journal of Industrial and Management Optimization*, 9 (2013), 595–619.

[13] Y. Narushima, H. Yabe and J. A. Ford, A three-term conjugate gradient method with sufficient descent property for unconstrained optimization, *SIAM Journal on Optimization*, 21 (2011), 212–230.

[14] Q. Ni, A globally convergent method of moving asymptotes with trust region technique, *Optimization methods and software*, 18 (2003), 283–297.

[15] E. Polak and G. Ribiere, Note sur la convergence de méthodes de directions conjuguées, *Revue française d’informatique et de recherche opérationnelle, série rouge*, 3 (1969), 35–43.

[16] B. T. Polyak, The conjugate gradient method in extremal problems, *USSR Computational Mathematics and Mathematical Physics*, 9 (1969), 94–112.

[17] K. Svanberg, The method of moving asymptotes—a new method for structural optimization, *International Journal for Numerical Methods in Engineering*, 24 (1987), 359–373.

[18] H. Wang and Q. Ni, A new method of moving asymptotes for large-scale unconstrained optimization, *Applied Mathematics and Computation*, 203 (2008), 62–71.
[19] W. Zhou and Y. Zhou, On the strong convergence of a modified Hestenes-Stiefel method for nonconvex optimization, *Journal of Industrial and Management Optimization*, 9 (2013), 893–899.

Received January 2015; 1st revision July 2015; 2nd revision December 2015.

E-mail address: 163zgh8163.com
E-mail address: niqfs@nuaa.edu.cn
E-mail address: zml571@126.com