J Optim Theory Appl (2015) 165:209–224
DOI 10.1007/s10957-014-0528-4

A Modified Self-Scaling Memoryless Broyden–Fletcher–Goldfarb–Shanno Method for Unconstrained Optimization

C. X. Kou · Y. H. Dai

Received: 7 January 2013 / Accepted: 18 January 2014 / Published online: 18 March 2014

Abstract The introduction of quasi-Newton and nonlinear conjugate gradient methods revolutionized the field of nonlinear optimization. The self-scaling memoryless Broyden–Fletcher–Goldfarb–Shanno (SSML-BFGS) method by Perry (Disscussion Paper 269, 1977) and Shanno (SIAM J Numer Anal, 15, 1247–1257, 1978) provided a good understanding about the relationship between the two classes of methods. Based on the SSML-BFGS method, new conjugate gradient algorithms, called CG_DESCENT and CGOPT, have been proposed by Hager and Zhang (SIAM J Optim, 16, 170–192, 2005) and Dai and Kou (SIAM J Optim, 23, 296–320, 2013), respectively. It is somewhat surprising that the two conjugate gradient methods perform more efficiently than the SSML-BFGS method. In this paper, we aim at proposing some suitable modifications of the SSML-BFGS method such that the sufficient descent condition holds. Convergence analysis of the modified method is made for convex and nonconvex functions, respectively. The numerical experiments for the testing problems from the Constrained and Unconstrained Test Environment collection demonstrate that the modified SSML-BFGS method yields a desirable improvement over CGOPT and the original SSML-BFGS method.

Communicated by N. Yamashita.

C. X. Kou
School of Science, Beijing University of Posts and Telecommunications, Beijing 100876, People's Republic of China
e-mail: koucx@bupt.edu.cn

Y. H. Dai (✉)
LSEC, ICMSEC, Academy of Mathematics and Systems Science, Chinese Academy of Sciences, Box 2719, Beijing 100190, People's Republic of China
e-mail: dyh@lsec.cc.ac.cn

Keywords Unconstrained optimization · Self-scaling · Conjugate gradient method · Quasi-Newton method · Global convergence · Improved Wolfe line search

Mathematics Subject Classification (2000) 49M37 · 90C30

1 Introduction

It is well known that the introduction of quasi-Newton and nonlinear conjugate gradient methods revolutionized the field of nonlinear optimization. The self-scaling memoryless Broyden–Fletcher–Goldfarb–Shanno (SSML-BFGS) method by Perry [1] and Shanno [2] provided a good understanding about the relationship between nonlinear conjugate gradient methods and quasi-Newton methods. Specifically, if the line search is exact and the identity matrix is used for the initial Hessian approximation, then both the BFGS and SSML-BFGS methods will generate the same iterations as the conjugate gradient method does for convex quadratic functions. On the other hand, if the previous Hessian approximation is utilized for updating the current quasi-Newton matrix in the SSML-BFGS method, it will become the self-scaling BFGS method.

Recently, based on the SSML-BFGS method, two efficient conjugate gradient algorithms, called CG_DESCENT and CGOPT, have been proposed by Hager and Zhang [3] and Dai and Kou [4], respectively. The numerical experiments show that, equipped with some nonmonotone line searches, both CG_DESCENT and CGOPT perform more efficiently than the SSML-BFGS method. This is somewhat surprising since the SSML-BFGS method can be regarded as a three-term conjugate gradient method, and three-term conjugate gradient methods (like the Beale-Powell restart method by Beale [5] and Powell [6]) were generally believed to outperform two-term conjugate gradient methods in practical computations.

This paper will focus on the original SSML-BFGS method with slight modifications so that the sufficient descent condition (see (12)) can be satisfied at each iteration. One possibility is to consider some modified secant equation (see, for example, [7–9]) to meet the sufficient descent condition. Following this line, however, we failed to find a better alternative after considerable numerical efforts. We shall explore two other possibilities in Sect. 3. It turned out that both of them are quite efficient in numerical tests.

The rest of this paper is organized as follows. Some preliminaries are made in Sect. 2. The modified SSML-BFGS method is provided in Sect. 3. Section 4 provides global convergence analysis for convex and nonconvex objective functions, respectively. Numerical results are reported in Sect. 5. Conclusions and discussions are made in the last section.

2 Preliminaries

Consider the unconstrained optimization problem

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

where f is smooth and its gradient $\mathbf{g}$ is available. More exactly, we assume that f satisfies

Assumption 2.1 (i) f is bounded below; namely, $f(\mathbf{x}) > -\infty$ for any $\mathbf{x} \in \mathbb{R}^n$; (ii) f is differentiable and its gradient $\mathbf{g}$ is Lipschitz continuous; namely, there exists a constant $L > 0$ such that

$$\|\mathbf{g}(\mathbf{x}) - \mathbf{g}(\mathbf{y})\| \leq L\|\mathbf{x} - \mathbf{y}\|, \quad \forall\, \mathbf{x}, \mathbf{y} \in \mathbb{R}^n, \tag{1}$$

where $\|\cdot\|$ stands for the Euclidean norm.

The SSML-BFGS method by Perry [1] and Shanno [2] is of the form

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{d}_k, \tag{2}$$

where the stepsize $\alpha_k > 0$ is obtained by some line search. The next search direction $\mathbf{d}_{k+1}$ ($k \geq 1$) is generated by

$$\mathbf{d}_{k+1} = -\mathbf{H}_{k+1}\, \mathbf{g}_{k+1}, \tag{3}$$

where the approximation matrix $\mathbf{H}_{k+1}$ is obtained by the BFGS update from a scaled identity matrix $\frac{1}{\tau_k}\mathbf{I}$. More exactly,

$$\mathbf{H}_{k+1} = \frac{1}{\tau_k}\left(\mathbf{I} - \frac{\mathbf{s}_k\mathbf{y}_k^T + \mathbf{y}_k\mathbf{s}_k^T}{\mathbf{s}_k^T\mathbf{y}_k}\right) + \left(1 + \frac{1}{\tau_k}\frac{\|\mathbf{y}_k\|^2}{\mathbf{s}_k^T\mathbf{y}_k}\right)\frac{\mathbf{s}_k\mathbf{s}_k^T}{\mathbf{s}_k^T\mathbf{y}_k}, \tag{4}$$

where $\mathbf{s}_k = \mathbf{x}_{k+1} - \mathbf{x}_k$ and $\mathbf{y}_k = \mathbf{g}_{k+1} - \mathbf{g}_k$. The scaling parameter τ_k in (4) is suggested [10,11] to lie in the interval

$$\tau_k \in \left[\frac{\mathbf{s}_k^T\mathbf{y}_k}{\|\mathbf{s}_k\|^2}, \frac{\|\mathbf{y}_k\|^2}{\mathbf{s}_k^T\mathbf{y}_k}\right]. \tag{5}$$

If the line search is exact, the SSML-BFGS search direction reduces to the conjugate gradient direction, whereas if we update $\mathbf{H}_{k+1}$ from $\frac{1}{\tau_k}\mathbf{H}_k$, the SSML-BFGS search direction will become the self-scaling BFGS direction.

By substituting (4) into (3), we can rewrite the search direction of the SSML-BFGS method of Perry [1] and Shanno [2] (with a multiplier difference) as

$$\mathbf{d}_{k+1}^{PS} = -\mathbf{g}_{k+1} + \left[\frac{\mathbf{g}_{k+1}^T\mathbf{y}_k}{\mathbf{s}_k^T\mathbf{y}_k} - \left(\tau_k + \frac{\|\mathbf{y}_k\|^2}{\mathbf{s}_k^T\mathbf{y}_k}\right)\frac{\mathbf{g}_{k+1}^T\mathbf{s}_k}{\mathbf{s}_k^T\mathbf{y}_k}\right]\mathbf{s}_k + \frac{\mathbf{g}_{k+1}^T\mathbf{s}_k}{\mathbf{s}_k^T\mathbf{y}_k}\mathbf{y}_k. \tag{6}$$

Based on this method, new conjugate gradient algorithms, called CG_DESCENT and CGOPT, have been proposed by Hager and Zhang [3] and Dai and Kou [4], respectively. Specifically, CG_DESCENT is derived by deleting the last term in (6) and setting the parameter τ_k to be the value $\frac{\|\mathbf{y}_k\|^2}{\mathbf{s}_k^T\mathbf{y}_k}$. CGOPT is proposed by seeking the vector on the manifold $\mathcal{S}_{k+1} = \{-\mathbf{g}_{k+1} + \beta \mathbf{s}_k : \beta \in \mathbb{R}\}$ that is closest to $\mathbf{d}_{k+1}^{PS}$ in (6) and setting $\tau_k = \frac{\mathbf{s}_k^T \mathbf{y}_k}{\|\mathbf{s}_k\|^2}$. More exactly, CGOPT is based on the following scheme

$$\mathbf{d}_{k+1} = -\mathbf{g}_{k+1} + \beta_k \mathbf{d}_k, \tag{7}$$

$$\beta_k = \frac{\mathbf{g}_{k+1}^T \mathbf{y}_k}{\mathbf{d}_k^T \mathbf{y}_k} - \frac{\mathbf{d}_k^T \mathbf{g}_{k+1}}{\mathbf{d}_k^T \mathbf{y}_k} \frac{\|\mathbf{y}_k\|^2}{\mathbf{d}_k^T \mathbf{y}_k} \tag{8}$$

(here we should notice that the formula (7) has also been used in CG_DESCENT since its version 5.1). The numerical experiments showed that the two conjugate gradient methods both perform more efficiently than the SSML-BFGS method in (6).

It is worth mentioning that CGOPT is equipped with the so-called improved Wolfe line search [4]. This line search consists in calculating a stepsize satisfying the following conditions

$$\phi_k(\alpha) \le \phi_k(0) + \min\left\{\epsilon |\phi_k(0)|,\ \delta \alpha \phi_k'(0) + \eta_k\right\}, \tag{9}$$

$$\phi_k'(\alpha) \ge \sigma\, \phi_k'(0). \tag{10}$$

Under the above conditions, the suggested value of the parameters are $\epsilon = 10^{-6}$, $\delta = 0.1$, $\sigma = 0.9$, and $\eta_k = 1/k^2$. The reason why we choose this line search is that, it not only performs well in numerical experiments, but also enables the Zoutendijk condition (11) (see [12]), which plays a basic role in the global convergence analysis. Further, CGOPT with the improved Wolfe line search is shown to be globally convergent for nonconvex functions in [4]. Throughout this paper, the line search is assumed to be the above improved Wolfe line search.

Lemma 2.1 *Assume that f satisfies Assumption 2.1. Consider the iterative method of the form* (2) *where the direction $\mathbf{d}_k$ satisfies $\mathbf{g}_k^T \mathbf{d}_k < 0$ and the stepsize α_k satisfies* (9) *and* (10)*. Then, we have that*

$$\sum_{k \ge 1} \frac{(\mathbf{g}_k^T \mathbf{d}_k)^2}{\|\mathbf{d}_k\|^2} < \infty. \tag{11}$$

3 The Modified SSML-BFGS Method

As can easily be seen, the SSML-BFGS method can be regarded as a certain three-term conjugate gradient method since it reduces to the linear conjugate gradient method if the objective function is quadratic and if the line search is exact. The third term in (6) is vanished in the ideal case, but can keep more information for nonlinear functions and inexact line searches. On the other hand, the SSML-BFGS method keeps some good properties of the BFGS method like the least change property. So it is reasonable to expect that the SSML-BFGS method performs better than the two-term conjugate gradient method in numerical computations. But the existing numerical results showed that the conjugate gradient algorithms, CG_DESCENT and CGOPT, both perform more efficiently than the SSML-BFGS method in (6). The puzzle attracts us to study the SSML-BFGS method with as light modifications as possible and with numerical performances better than CGOPT.

We shall focus on the original SSML-BFGS method with slight modifications so that the sufficient descent condition can be satisfied at each iteration; namely,

$$\mathbf{g}_k^T \mathbf{d}_k \leq -c\|\mathbf{g}_k\|^2, \quad \text{for all } k \geq 1 \text{and some constant } c > 0. \tag{12}$$

One possibility is to consider some modified secant equation (see, for example, [7–9])

$$\mathbf{B}_{k+1}\mathbf{s}_k = \mathbf{z}_k, \qquad \mathbf{z}_k = \mathbf{y}_k + \lambda \frac{\theta_k}{\|\mathbf{s}_k\|^2}\mathbf{s}_k,$$

where $\theta_k = 6(f_k - f_{k+1}) + 3(\mathbf{g}_k + \mathbf{g}_{k+1})^T\mathbf{s}_k$ and $\lambda \geq 0$ is some constant. By replacing the vector $\mathbf{y}_k$ in the SSML-BFGS method with $\mathbf{z}_k$, it is not difficult to show that the method satisfies the sufficient descent condition in a way similar to the related references, and better numerical performance is expected. Following this line, however, we failed to find an alternative better than the original SSML-BFGS method after considerable numerical efforts.

For other possibilities, we consider the following modification and multiply the third term in (6) by some non-negative parameter ξ_k, yielding the following direction

$$\mathbf{d}_{k+1} = -\mathbf{g}_{k+1} + \beta_k(\tau_k)\mathbf{d}_k + \gamma_k\mathbf{y}_k, \tag{13}$$

where

$$\beta_k(\tau_k) = \frac{\mathbf{g}_{k+1}^T\mathbf{y}_k}{\mathbf{d}_k^T\mathbf{y}_k} - \left(\tau_k + \frac{\|\mathbf{y}_k\|^2}{\mathbf{s}_k^T\mathbf{y}_k}\right)\frac{\mathbf{g}_{k+1}^T\mathbf{s}_k}{\mathbf{d}_k^T\mathbf{y}_k}, \tag{14}$$

$$\gamma_k = \xi_k \frac{\mathbf{g}_{k+1}^T\mathbf{d}_k}{\mathbf{d}_k^T\mathbf{y}_k}, \quad 0 \leq \xi_k \leq 1 \tag{15}$$

If $\mathbf{d}_k^T\mathbf{g}_{k+1} = 0$, the scalar $\beta_k(\tau_k)$ in (14) reduces to the Hestenes-Stiefel or Polak-Ribiére-Polyak formula since its second term is missing, and the parameter γ_k in (15) reduces to zero. Powell [13] constructed a counter-example showing that the Polak-Ribiére-Polyak method with exact line searches may not converge for general nonlinear functions.

Consequently, Powell's example can also be used to show that the method (2) and (13) with $\beta_k(\tau_k)$ and γ_k given by (14) and (15) need not converge for general nonlinear functions. Therefore, we replace (14)s by the following truncation form

$$\beta_k^+(\tau_k) = \max\left\{\beta_k(\tau_k),\ \zeta\frac{\mathbf{g}_{k+1}^T\mathbf{d}_k}{\|\mathbf{d}_k\|^2}\right\}, \tag{16}$$

where $0 < \zeta < 1$ is some parameter and its suggested value is 0.1 in our practical computations. This kind of truncation form comes from a similar idea in [4]. In this case, if β_k is truncated; namely, $\beta_k^+(\tau_k) = \zeta \frac{\mathbf{g}_{k+1}^T \mathbf{d}_k}{\|\mathbf{d}_k\|^2}$, we set $\xi_k \equiv 0$ in (15) to guarantee the sufficient descent condition. In other words, if a truncation happens, we restart the method along the direction

$$\mathbf{d}_{k+1} = -\mathbf{g}_{k+1} + \zeta \frac{\mathbf{g}_{k+1}^T \mathbf{d}_k}{\|\mathbf{d}_k\|^2} \mathbf{d}_k. \tag{17}$$

The above direction reduces to the steepest descent direction $-\mathbf{g}_{k+1}$ if the line search is exact. However, they are not the same if the line search is not exact, which is the usual case in practical computations. Since the direction $\mathbf{d}_{k+1}$ makes use of some derivative information along $\mathbf{d}_k$, it is reasonable to believe that the restart direction $\mathbf{d}_{k+1}$ in restart is better than $-\mathbf{g}_{k+1}$ in restart. This is proved by our numerical tests.

Now we describe the modified SSML-BFGS method as follows.

Algorithm 3.1 ***(Modified SSML-BFGS method)***

Step 0. Given $\mathbf{x}_1 \in \mathbb{R}^n$, $\varepsilon \geq 0$ *and* $0 < \zeta < 1$.
Step 1. Set $k := 1$. *If* $\|\mathbf{g}_1\| \leq \varepsilon$, *stop. Let* $\mathbf{d}_1 = -\mathbf{g}_1$.
Step 2. Compute a stepsize $\alpha_k > 0$ *satisfying conditions (9), (10).*
Step 3. Let $\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{d}_k$. *Compute* $\mathbf{g}_{k+1}$. *If* $\|\mathbf{g}_{k+1}\| \leq \varepsilon$, *stop.*
Step 4. If the dynamic restart conditions are satisfied, let $\mathbf{d}_{k+1} = -\mathbf{g}_{k+1}$ *and set* $k := k + 1$, *goto Step 2.*
Step 5. Compute β_k *by (14) and* β_k^+ *by (16). If* $\beta_k^+ = \beta_k$, *decide* $\xi_k \in [0, 1]$ *and compute* γ_k *by (15) and* $\mathbf{d}_{k+1}$ *by (13); else, update* $\mathbf{d}_{k+1} = -\mathbf{g}_{k+1} + \beta_k^+ \mathbf{d}_k$. *Set* $k := k + 1$, *goto Step 2.*

Step 2 of the algorithm means that the stepsizes are obtained by the improved Wolfe line search algorithm (see Algorithm 3.2 in [4] for details). It is a nonmonotone line search algorithm which can not only avoid a numerical drawback of the Wolfe line search, but also guarantee the global convergence of conjugate gradient methods. In Step 4, to accelerate the algorithm, we adopt the dynamic restart technique proposed in [4] (see Algorithm 4.1 in [4] for details).

In Step 5, we need to give the value of $\xi_k \in [0, 1]$, when the parameter β_k need not be truncated by β_k^+. Here, we propose two candidates for ξ_k:

(i) given some constant $\xi \in 0 \leq \xi < 1$, set $\xi_k \equiv \xi$.
(ii) given $\xi \in [0, 1]$ and a positive constant c_0, calculate $\bar{\xi}_k$ by the following way

$$\bar{\xi}_k = \arg\max \left\{ \xi_k \leq 1 \,|\, \mathbf{d}_{k+1}^T \mathbf{g}_{k+1} \leq -c_0 \|\mathbf{g}_{k+1}\|^2 \right\}. \tag{18}$$

If $\bar{\xi}_k \in 0 \leq \bar{\xi}_k < 1$ and $\bar{\xi}_k > \xi$, we set $\xi_k = \bar{\xi}_k$; else set $\xi_k = \xi$.

In (i), by letting ξ_k to be a constant ξ in [0, 1], we can show that the direction (13)–(15) satisfies the sufficient descent condition in the following lemmas. In (ii), we propose a dynamic way to determine ξ_k. With such ξ_k, the direction (13)–(15) not only satisfies the sufficient descent condition, but also is as close as possible to the SSML-BFGS direction.

Now, we can establish the sufficient descent condition for Algorithm 3.1.

Lemma 3.1 *Consider the conjugate gradient method (2), (13) with (14) and (15). If* $\mathbf{d}_k^T \mathbf{y}_k > 0$*, we always have that*

$$\mathbf{d}_{k+1}^T \mathbf{g}_{k+1} \leq -c_1 \|\mathbf{g}_{k+1}\|^2 \text{ for some positive constant } c_1 > 0. \tag{19}$$

Proof Based on the choice of ξ_k, we divided the proof into the following two cases.

Case (i). From (13)–(15) and the positivity of $\mathbf{s}_k^T \mathbf{y}_k$ and τ_k, it follows that

$$\begin{aligned}\mathbf{d}_{k+1}^T \mathbf{g}_{k+1} &= \frac{-(\mathbf{s}_k^T \mathbf{y}_k)^2 \|\mathbf{g}_{k+1}\|^2 - (\mathbf{s}_k^T \mathbf{g}_{k+1})^2 \|\mathbf{y}_k\|^2 + (1+\xi)(\mathbf{s}_k^T \mathbf{g}_{k+1})(\mathbf{s}_k^T \mathbf{y}_k)\mathbf{g}_{k+1}^T \mathbf{y}_k}{(\mathbf{s}_k^T \mathbf{y}_k)^2} \\ &\quad -\tau_k \frac{(\mathbf{s}_k^T \mathbf{g}_{k+1})^2}{\mathbf{s}_k^T \mathbf{y}_k} \\ &\leq \frac{-(\mathbf{s}_k^T \mathbf{y}_k)^2 \|\mathbf{g}_{k+1}\|^2 - (\mathbf{s}_k^T \mathbf{g}_{k+1})^2 \|\mathbf{y}_k\|^2 + (1+\xi)(\mathbf{s}_k^T \mathbf{g}_{k+1})(\mathbf{s}_k^T \mathbf{y}_k)\mathbf{g}_{k+1}^T \mathbf{y}_k}{(\mathbf{s}_k^T \mathbf{y}_k)^2}.\end{aligned} \tag{20}$$

By applying the inequality

$$\mathbf{u}^T \mathbf{v} \leq \frac{1}{2}(\|\mathbf{u}\|^2 + \|\mathbf{v}\|^2)$$

to the relation (20) with

$$\mathbf{u} = \frac{1+\xi}{\sqrt{2}}(\mathbf{s}_k^T \mathbf{y}_k)\mathbf{g}_{k+1}, \quad \mathbf{v} = \sqrt{2}(\mathbf{s}_k^T \mathbf{g}_{k+1})\mathbf{y}_k,$$

we can obtain

$$\mathbf{d}_{k+1}^T \mathbf{g}_{k+1} \leq -\left(1 - \frac{(1+\xi)^2}{4}\right) \|\mathbf{g}_{k+1}\|^2.$$

Thus, (19) holds with $c_1 = 1 - \frac{(1+\xi)^2}{4}$.

Case (ii). From Case (i) and (18), it is easy to see that (19) holds with $c_1 = \min\{c_0, 1 - \frac{(1+\xi)^2}{4}\}$.

Lemma 3.2 *Consider the conjugate gradient method (2) and (13), where* $\beta_k(\tau_k)$ *is replaced by* $\beta_k^+(\tau_k)$ *in (16) and* γ_k *is given in (15). If* $\mathbf{d}_k^T \mathbf{y}_k > 0$ *and the line search satisfies condition (10), we have that*

$$\mathbf{d}_{k+1}^T \mathbf{g}_{k+1} \leq -c_2 \|\mathbf{g}_{k+1}\|^2 \text{ for some positive constant } c_2 > 0. \tag{21}$$

Proof By Algorithm 3.1 and Lemma 3.1, we only need to consider the case that

$$\beta_k^+ = \zeta \frac{\mathbf{g}_{k+1}^T \mathbf{d}_k}{\|\mathbf{d}_k\|^2} \text{ with } 0 < \zeta < 1 \text{ and } \xi_k \equiv 0 \text{ in (15).}$$

In this case, it is obvious that

$$\begin{aligned} \mathbf{d}_{k+1}^T \mathbf{g}_{k+1} &= -\|\mathbf{g}_{k+1}\|^2 + \zeta \frac{(\mathbf{d}_k^T \mathbf{g}_{k+1})^2}{\|\mathbf{d}_k\|^2} \\ &\le -(1-\zeta)\,\|\mathbf{g}_{k+1}\|^2. \end{aligned}$$

This, with Lemma 3.1, indicates that (21) holds with $c_2 = \min\{c_1,\ (1-\zeta)\}$. □

4 Global Convergence Analysis

For uniformly convex functions, we have the following convergence result.

Theorem 4.1 *Assume that f satisfies Assumption 2.1. Consider the search direction defined by (13)–(15), where the parameter τ_k lies in the interval in (5) and the stepsize α_k is calculated by the improved line search satisfying (9) and (10). If, further, f is uniformly convex; namely, there exists a constant $\mu > 0$ such that*

$$(\nabla f(\mathbf{x}) - \nabla f(\mathbf{y}))^T (\mathbf{x}-\mathbf{y}) \ge \mu \|\mathbf{x}-\mathbf{y}\|^2, \quad \forall\, \mathbf{x}, \mathbf{y} \in \mathbb{R}^n, \tag{22}$$

we have that

$$\lim_{k\to\infty} \mathbf{g}_k = 0. \tag{23}$$

Proof It follows from (1) and (22) that

$$\|\mathbf{y}_k\| \le L\,\|\mathbf{s}_k\|, \tag{24}$$

$$\mathbf{d}_k^T \mathbf{y}_k \ge \mu\,\|\mathbf{d}_k\|\,\|\mathbf{s}_k\|. \tag{25}$$

By (22) and (24), it is easy to see that for any τ_k belonging to the interval in (5), there exists a positive constant c_τ such that

$$|\tau_k| \le c_\tau.$$

Consequently,

$$\begin{aligned} \|\mathbf{d}_{k+1}\| &\le \|\mathbf{g}_{k+1}\| + \left| \frac{\mathbf{g}_{k+1}^T \mathbf{y}_k}{\mathbf{d}_k^T \mathbf{y}_k} - \left(\tau_k + \frac{\|\mathbf{y}_k\|^2}{\mathbf{s}_k^T \mathbf{y}_k}\right) \frac{\mathbf{g}_{k+1}^T \mathbf{s}_k}{\mathbf{d}_k^T \mathbf{y}_k} \right| \|\mathbf{d}_k\| + \left| \xi_k \frac{\mathbf{g}_{k+1}^T \mathbf{d}_k}{\mathbf{d}_k^T \mathbf{y}_k} \right| \|\mathbf{y}_k\| \\ &\le \left(1 + \frac{L\,\|\mathbf{s}_k\|\,\|\mathbf{d}_k\|}{\mathbf{d}_k^T \mathbf{y}_k} + (c_\tau + \frac{L^2}{\mu}) \frac{\|\mathbf{s}_k\|\,\|\mathbf{d}_k\|}{\mathbf{d}_k^T \mathbf{y}_k} + \frac{\|\mathbf{d}_k\|\,\|\mathbf{y}_k\|}{\mathbf{d}_k^T \mathbf{y}_k}\right) \|\mathbf{g}_{k+1}\| \\ &\le \left(1 + \frac{L^2 + 2\mu L + \mu c_\tau}{\mu^2}\right) \|\mathbf{g}_{k+1}\|. \end{aligned} \tag{26}$$

On the other hand, Lemmas 3.1 and 2.1 imply that

$$\sum_{k\geq 1} \frac{\|\mathbf{g}_k\|^4}{\|\mathbf{d}_k\|^2} < \infty. \tag{27}$$

Therefore, we obtain by (26) and (27) that

$$\sum_{k\geq 1} \|\mathbf{g}_k\|^2 < \infty,$$

which implies the truth of (23).

For general nonlinear functions, similar to [14] and [15], we can establish a weaker convergence result in the sense that

$$\liminf_{k\to\infty} \|\mathbf{g}_k\| = 0. \tag{28}$$

To this aim, we proceed by contradiction and assuming that there exists $\gamma > 0$ such that

$$\|\mathbf{g}_k\| \geq \gamma, \quad \forall\, k \geq 1. \tag{29}$$

Lemma 4.1 *Assume that f satisfies Assumption 2.1. Consider the iterative method of the form (2), where the search direction d_k is defined by (13), (14), (16), and (15) with k replaced with $k+1$, and where the stepsize α_k is calculated by the improved Wolfe line search satisfying (9) and (10). If the generated sequence $\{\mathbf{x}_k\}$ is bounded and (29) holds, we have that $\mathbf{d}_k \neq 0$ and*

$$\sum_{k\geq 2} \|\mathbf{u}_k - \mathbf{u}_{k-1}\|^2 < \infty, \tag{30}$$

where $\mathbf{u}_k = \mathbf{d}_k/\|\mathbf{d}_k\|$.

Proof First, note that $\mathbf{d}_k \neq 0$, for otherwise the sufficient descent condition (21) would imply $\mathbf{g}_k = 0$. Therefore, $\mathbf{u}_k$ is well defined. Now, divide formula (16) for β_k^+ into two parts as follows

$$\beta_k^{(1)} = \max\left\{ \frac{\mathbf{g}_{k+1}^T\mathbf{y}_k}{\mathbf{d}_k^T\mathbf{y}_k} - \left(\tau_k + \frac{\|\mathbf{y}_k\|^2}{\mathbf{s}_k^T\mathbf{y}_k}\right)\frac{\mathbf{g}_{k+1}^T\mathbf{s}_k}{\mathbf{d}_k^T\mathbf{y}_k} - \zeta\frac{\mathbf{d}_k^T\mathbf{g}_{k+1}}{\|\mathbf{d}_k\|^2},\ 0 \right\}, \tag{31}$$

$$\beta_k^{(2)} = \zeta\frac{\mathbf{d}_k^T\mathbf{g}_{k+1}}{\|\mathbf{d}_k\|^2} \quad (0 < \zeta < 1) \tag{32}$$

and define

$$\mathbf{w}_k = \frac{-\mathbf{g}_k + \beta_{k-1}^{(2)}\mathbf{d}_{k-1} + \gamma_{k-1}\mathbf{y}_{k-1}}{\|\mathbf{d}_k\|} \quad \text{and} \quad \delta_k = \frac{\beta_{k-1}^{(1)}\|\mathbf{d}_{k-1}\|}{\|\mathbf{d}_k\|}.$$

By $\mathbf{d}_k = -\mathbf{g}_k + \beta_{k-1}^{+}\mathbf{d}_{k-1} + \gamma_k \mathbf{y}_{k-1}$, we have for $k \geq 2$,

$$\mathbf{u}_k = \mathbf{w}_k + \delta_k \mathbf{u}_{k-1}. \tag{33}$$

Using the identity $\|\mathbf{u}_k\| = \|\mathbf{u}_{k-1}\| = 1$ and (33), we obtain

$$\|\mathbf{w}_k\| = \|\mathbf{u}_k - \delta_k \mathbf{u}_{k-1}\| = \|\delta_k \mathbf{u}_k - \mathbf{u}_{k-1}\| \tag{34}$$

(the last equality can be verified by squaring both sides). Using the condition that $\delta_k \geq 0$, the triangle inequality and (34), we have that

$$\begin{aligned}\|\mathbf{u}_k - \mathbf{u}_{k-1}\| &\leq \|(1+\delta_k)\mathbf{u}_k - (1+\delta_k)\mathbf{u}_{k-1}\| \\ &\leq \|\mathbf{u}_k - \delta_k \mathbf{u}_{k-1}\| + \|\delta_k \mathbf{u}_k - \mathbf{u}_{k-1}\| \\ &= 2\|\mathbf{w}_k\|.\end{aligned} \tag{35}$$

The line search condition (10) indicates that

$$\mathbf{g}_{k+1}^T \mathbf{d}_k \geq \sigma \mathbf{g}_k^T \mathbf{d}_k, \tag{36}$$

which implies that

$$\mathbf{d}_k^T \mathbf{y}_k = \mathbf{g}_{k+1}^T \mathbf{d}_k - \mathbf{g}_k^T \mathbf{d}_k \geq -(1-\sigma)\mathbf{d}_k^T \mathbf{g}_k. \tag{37}$$

Combining (36) and (37), we get that

$$\mathbf{g}_{k+1}^T \mathbf{d}_k \geq \sigma \mathbf{g}_k^T \mathbf{d}_k \geq \frac{-\sigma}{1-\sigma}\mathbf{d}_k^T \mathbf{y}_k. \tag{38}$$

It follows from the equality in (37) and $\mathbf{d}_k^T \mathbf{g}_k < 0$ that

$$\mathbf{g}_{k+1}^T \mathbf{d}_k \leq \mathbf{d}_k^T \mathbf{y}_k. \tag{39}$$

Dividing $\mathbf{d}_k^T \mathbf{y}_k$ in both (38) and (39), we easily get for all $k \geq 1$, the parameter γ_k in (15) satisfies the following inequality

$$|\gamma_k| = \left| \xi_k \frac{\mathbf{d}_k^T \mathbf{g}_{k+1}}{\mathbf{d}_k^T \mathbf{y}_k} \right| \leq \max\{\frac{\sigma}{1-\sigma}, 1\},$$

which with the definition of $\beta_k^{(2)}$ in (31) gives

$$\| -\mathbf{g}_k + \beta_{k-1}^{(2)}\mathbf{d}_{k-1} + \gamma_{k-1}\mathbf{y}_{k-1}\| \leq (1+\zeta)\,\|\mathbf{g}_k\| + \max\{\frac{\sigma}{1-\sigma}, 1\}\|\mathbf{y}_{k-1}\|. \tag{40}$$

By the continuity of ∇f and the boundedness of $\{\mathbf{x}_k\}$, there exists some parameter constant $\bar{\gamma}$ such that

$$\|\mathbf{x}_k\| \le \bar{\gamma}, \quad \|\mathbf{g}_k\| \le \bar{\gamma}, \qquad \forall k \ge 1.$$

It follows from this and (40) that there exists some positive parameter C such that

$$\| -\mathbf{g}_k + \beta_{k-1}^{(2)}\mathbf{d}_{k-1} + \gamma_{k-1}\mathbf{y}_{k-1}\| \le C.$$

This bound for the numerator of $\mathbf{w}_k$ coupled with (35) gives

$$\|\mathbf{u}_k - \mathbf{u}_{k-1}\| \le 2\|\mathbf{w}_k\| \le \frac{2C}{\|\mathbf{d}_k\|}. \tag{41}$$

The relation (29), the sufficient descent condition (21), and the Zoutendijk condition (11) indicate that

$$\sum_{k\ge 1}\frac{1}{\|\mathbf{d}_k\|^2} \le \frac{1}{\gamma^4}\sum_{k\ge 1}\frac{\|\mathbf{g}_k\|^4}{\|\mathbf{d}_k\|^2} \le \frac{1}{\gamma^4 c_2^2}\sum_{k\ge 1}\frac{(\mathbf{g}_k^T\mathbf{d}_k)^2}{\|\mathbf{d}_k\|^2} < +\infty. \tag{42}$$

Thus, (30) follows from (41) and (42). □

By an argument similar to that in [14] and [4], we can obtain the following convergence theorem for general nonconvex functions. If the objective function has bounded level sets, it is easy to see that the generated sequence $\{\mathbf{x}_k\}$ is bounded.

Theorem 4.2 *Assume that f satisfies Assumption 2.1. Consider the iterative method of the form (2), where the search direction $\mathbf{d}_k$ is defined by (13), (14), (16), and (15) with k replaced with $k+1$, and where the stepsize α_k is calculated by the improved Wolfe line search satisfying (9) and (10). If the generated sequence $\{\mathbf{x}_k\}$ is bounded, the method converges in the sense that (28) holds.*

5 Numerical Experiments

In this section, we report the numerical results of our modified SSML-BFGS method of the form (2), (13)–(15), where the scaling parameter τ_k is chosen to be $\frac{\mathbf{s}_k^T\mathbf{y}_k}{\|\mathbf{s}_k\|^2}$. If the parameter ξ_k in (15) is determined by the two ways proposed in (i) and (ii) of Sect. 3, respectively, we call the corresponding variants of the modified SSML-BFGS method by *Algorithm 3.1(i)* and *Algorithm 3.1(ii)*. In all the compared algorithms, the stepsize α_k is calculated by the improved Wolfe line search [4] satisfying the conditions (9) and (10). We stop the iteration if the inequality

$$\|\mathbf{g}_k\|_2 \le 10^{-6}$$

is satisfied.

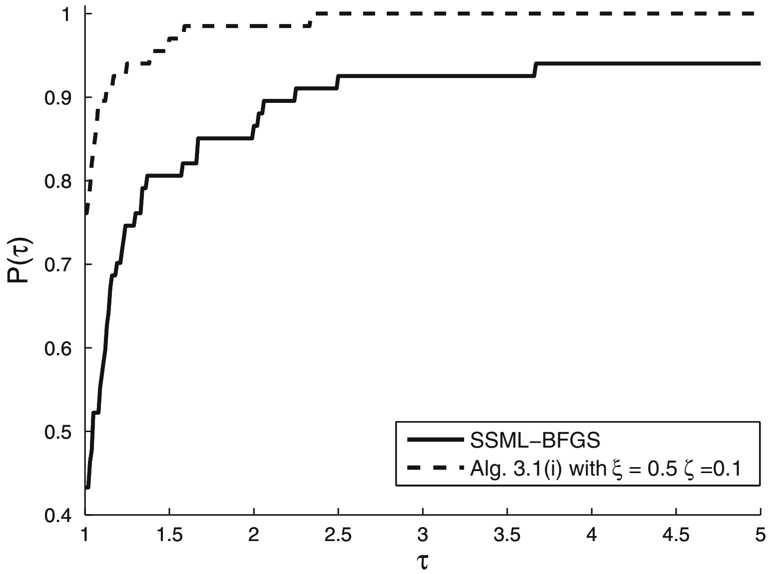

Fig. 1 Performance profile of Alg. 3.1(i) and SSML-BFGS based on CPU time for the whole test problems

The test problems are taken from the Constrained and Unconstrained Test Environment collection [16]. There are 118 unconstrained test problems in total and their dimensions vary from 50 to 10^4. For each comparison, however, we did not count those problems for which different solvers converge to different local minimizers.

The performance profile by Dolan and Moré [17] is used to display the performance of the algorithms. Define $\mathcal{P}$ as the whole set of n_p test problems and $\mathcal{S}$ the set of the interested solvers. Denote by $t_{p,s}$ the cpu time required by solver s for problem p. Define the performance ratio as

$$r_{p,s} = \frac{t_{p,s}}{t_p^*},$$

where $t_p^* = \min\{t_{p,s} : s \in \mathcal{S}\}$. It is obvious that $r_{p,s} \geq 1$ for all p and s. If a solver fails to solve a problem, then the ratio $r_{p,s}$ is assigned to be a large number 10^{10}. For each solver s, the performance profile is defined as the following cumulative distribution function for performance ratio $r_{p,s}$,

$$P(\tau) = \frac{size\{p \in \mathcal{P} : r_{p,s} \leq \tau\}}{n_p}.$$

That is, for each method, in the following figures, we plot the fraction $P(\tau)$ of problems for which the method is within a factor τ of the best time. Obviously, $P(1)$ represents the percentage of the test problems for which the method is the fastest. The top curve is the method that solved the most problems in a time that was within a factor τ of the best time. See [17] for more details about the performance profile.

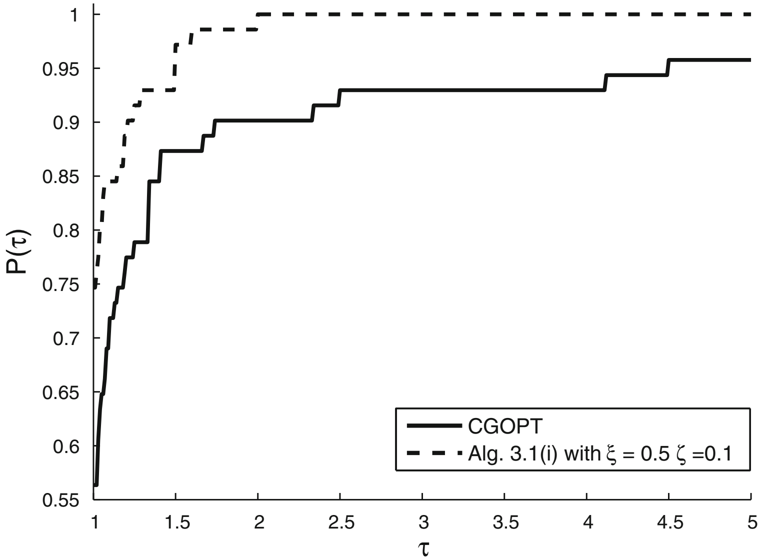

Fig. 2 Performance profile of Alg. 3.1(i) and CGOPT based on CPU time for hard test problems.

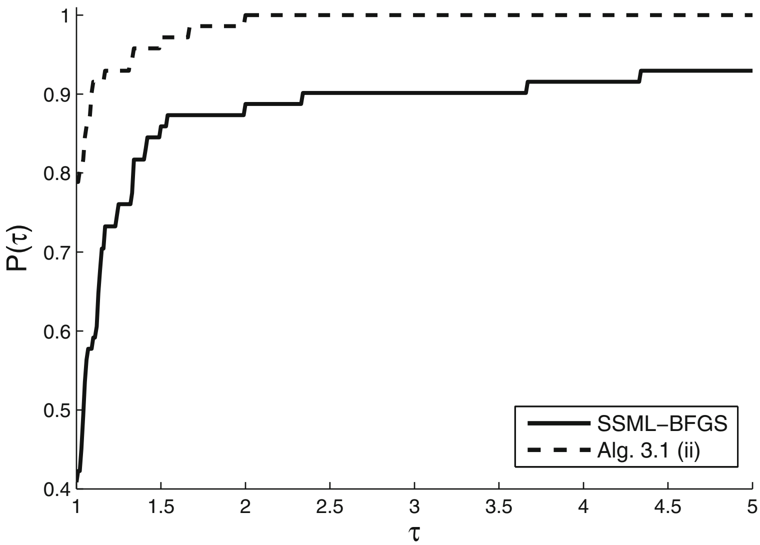

Fig. 3 Performance profile of Alg. 3.1(ii) and SSML-BFGS based on CPU time for the whole test problems.

In Fig. 1, we compare Algorithm 3.1 (i) and the original SSML-BFGS method. The two methods differ in that the former searches along the direction (13)–(15) and the latter does along the direction (6). After eliminating the problems for which the two variants converge to different local minimizers, 107 problems are left. Fig. 1 shows that the new method has a better performance than the SSML-BFGS method.

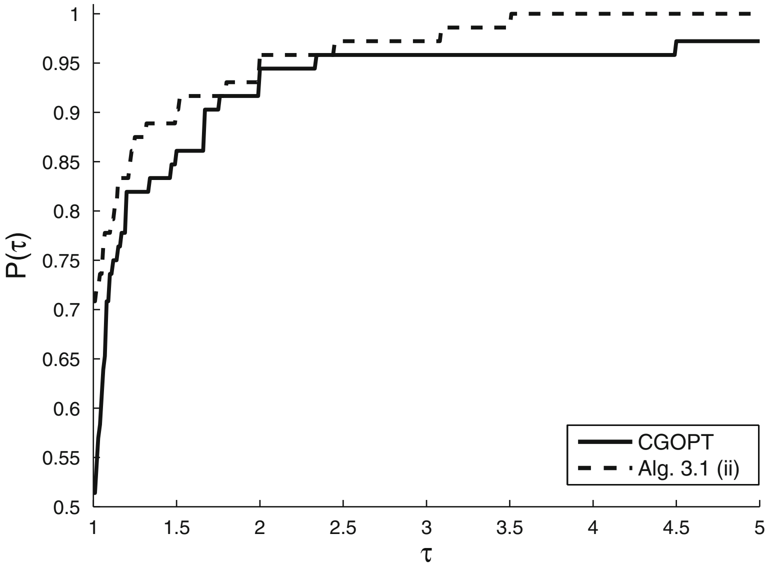

Fig. 4 Performance profile of Alg. 3.1(ii) and CGOPT based on CPU time for hard problems.

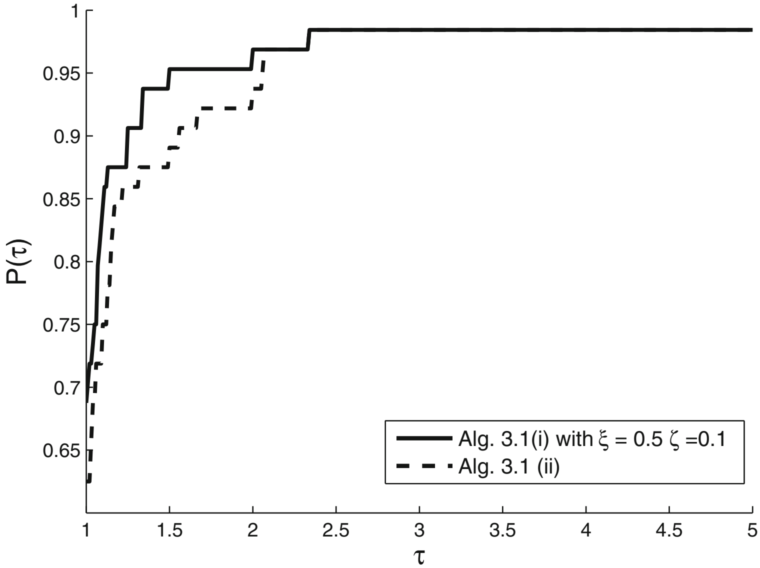

Fig. 5 Performance profile of Alg. 3.1(i) and (ii) based on CPU time for the whole test problems.

In Fig. 2, we compare Algorithm 3.1(i) with CGOPT [4] for those *hard* problems (here a test problem is said to be hard if any of the solvers requires at least 50 iterations regardless of the problem dimension). There are 70 hard problems. Fig. 2 shows that Algorithm 3.1(i) has a great advantage over CGOPT on the whole for the hard problems.

In Fig. 3, we compare Algorithm 3.1(ii) and the original SSML-BFGS method. After eliminating the problems for which the two variants converge to different local minimizers, 111 problems are left. Fig. 3 shows that Algorithm 3.1(ii) obtains a better performance than the SSML-BFGS method.

In Fig. 4, we compare Algorithm 3.1(ii) with CGOPT [4] for those hard problems (in other words, any of the solvers requires at least 50 iterations regardless of the problem dimension). There are 72 hard problems in this case. Fig. 4 shows that Algorithm 3.1(ii) has a great advantage over CGOPT on the whole for the hard problems.

At the end of this section, we present a comparison between Algorithm 3.1(i) and Algorithm 3.1(ii) for the whole test problems of CUTEr. From Fig. 5, we can see that Algorithm 3.1(i) and Algorithm 3.1(ii) perform nearly the same, except that Algorithm (3.1)(i) is slightly better when $\tau \leq 2.5$.

6 Conclusions

In this paper, we have proposed a modified SSML-BFGS method, which is of the form (2), (13), (16). The main modification to SSML-BFGS is that we multiply the third term of the SSML-BFGS direction with parameter ξ_k. We also proposed two strategies (i) and (ii) in Sect. 3 to determine this parameter. We proved that the modified SSML-BFGS method with either of the strategies satisfies the sufficient descent condition provided that $\mathbf{d}_k^T \mathbf{y}_k > 0$. Consequently, under some mild conditions, global convergence results have been established for convex and nonconvex objective functions, respectively. Numerical results indicated that the two implementations of the SSML-BFGS method perform better than the original SSML-BFGS method by Perry and Shanno. Moreover, they have a better performance than CGOPT for the hard problems from the CUTEr collection.

The two strategies for choosing the parameter ξ_k are proposed such that the direction generated by the modified SSML-BFGS method satisfies the sufficient descent condition (12). To some extent, this coincides with the same line along which nonlinear conjugate gradient methods develops. In a recent survey [18], nonlinear conjugate gradient methods are divided by the following three types. Early nonlinear conjugate gradient methods, including the Fletcher-Reeves, Polak-Ribiére-Polyak and Hestenes-Stiefel ones, may not generate a descent search direction even with strong Wolfe line searches. The second type is descent nonlinear conjugate gradient methods that can guarantee the descent property of the generated search direction. A remarkable representative is the Dai-Yuan method [19], whose descent property and global convergence can be achieved only with the Wolfe line search. Further, the hybridization of the Dai-Yuan and Hestenes-Stiefel methods can lead to an efficient conjugate gradient algorithm, called DYHS in [20]. The third type is nonlinear conjugate gradient methods that can guarantee the sufficient descent condition of the generated search direction such as the most efficient algorithms CG_DESCENT and CGOPT mentioned in this paper. Therefore, we feel that there might be a large room for improving numerical optimization methods with small storage, including the SSML-BFGS method and the three-term nonlinear conjugate gradient method, by imposing the sufficient descent condition.

Acknowledgments This work was partly supported by the Chinese NSF Grants (Nos. 11331012 and 81173633), the China National Funds for Distinguished Young Scientists (No. 11125107), and the CAS Program for Cross & Coorperative Team of the Science & Technology Innovation. The authors are very grateful to the two anonymous referees for their valuable suggestions and comments on an early version of this paper.

References

1. Perry, J. M.: A class of conjugate gradient algorithms with a two-step variable-metric memory. Discussion Paper 269, Center for Mathematical Studies in Economics and Management Sciences, Northwestern University, Evanston, Illinois (1977).
2. Shanno, D.F.: On the convergence of a new conjugate gradient algorithm. SIAM J. Numer. Anal. **15**, 1247–1257 (1978)
3. Hager, W.W., Zhang, H.: A new conjugate gradient method with guaranteed descent and an efficient line search. SIAM J. Optim. **16**, 170–192 (2005)
4. Dai, Y.H., Kou, C.X.: A nonlinear conjugate gradient algorithm with an optimal property and an improved Wolfe line search. SIAM J. Optim. **23**, 296–320 (2013)
5. Beale, E.M.L.: A derivation of conjugate gradients. In: Lootsman, F.A. (ed.) Numerical Methods for Nonlinear Optimization, pp. 39–43. Academic Press, London (1972)
6. Powell, M.J.D.: Restart procedures for the conjugate gradient method. Math. Progr. **12**, 241–254 (1977)
7. Wei, Z., Li, G., Qi, L.: New quasi-Newton methods for unconstrained optimization problems. Appl. Math. Comput. **175**, 1156–1188 (2006)
8. Zhang, J.Z., Deng, N.Y., Chen, L.H.: New quasi-Newton equation and related methods for unconstrained optimization. J. Optim. Theory Appl. **102**, 147–167 (1999)
9. Zhang, J.Z., Xu, C.X.: Properties and numerical performance of quasi-Newton methods with modified quasi-Newton equations. J. Comput. Appl. Math. **137**, 269–278 (2001)
10. Oren, S.S.: Self scaling variable metric (SSVM) algorithms, part II: implementation and experiments. Manag. Sci. **20**(5), 863–874 (1974)
11. Oren, S.S., Luenberger, D.G.: Self scaling variable metric (SSVM) algorithms, part I: criteria and sufficient conditions for scaling a class of algorithms. Manag. Sci. **20**(5), 845–862 (1974)
12. Zoutendijk, G.: Nonlinear programming, computational methods. In: Abadie, J. (ed.) Integer and Nonlinear Programming, pp. 37–86. North-Holland, Amsterdam (1970)
13. Powell, M.J.D.: Nonconvex minimization calculations and the conjugate gradient method. In: Griffiths, D.F. (ed.) Lecture Notes in Mathematics, pp. 122–141. Springer, Berlin (1984)
14. Gilbert, J.C., Nocedal, J.: Global convergence properties of conjugate gradient methods for optimization. SIAM J. Optim. **2**(1), 21–42 (1992)
15. Dai, Y.H., Liao, L.Z.: New conjugacy conditions and related nonlinear conjugate gradient methods. Appl. Math. Optim. **43**, 87–101 (2001)
16. Gould, N. I. M., Orban, D., Toint, Ph. L.: CUTEr (and SifDec), a constrained and unconstrained testing environment, revisited. Technical Report TR/PA/01/04, CERFACS, Toulouse, France (2001).
17. Dolan, E.D., Moré, J.J.: Benchmarking optimization software with performance profiles. Math. Progr. **91**, 201–213 (2002)
18. Dai, Y.H.: Nonlinear conjugate gradient methods. Published Online,Wiley Encyclopedia of Operations Research and Management Science(2011)
19. Dai, Y.H., Yuan, Y.: A nonlinear conjugate gradient with a strong global convergence property. SIAM J. Optim. **10**(1), 177–182 (1999)
20. Dai, Y.H., Yuan, Y.: An efficient hybrid conjugate gradient method for unconstrained optimization. Ann. Oper. Res. **103**, 33–47 (2001)