



Improved Hessian approximation with modified secant equations for symmetric rank-one method

Farzin Modarres*, Abu Hassan Malik, Wah June Leong

Department of Mathematics, Faculty of Science, Universiti Putra Malaysia, 43400 Serdang, Selangor, Malaysia

ARTICLE INFO

Article history:

Received 10 March 2009

Received in revised form 25 October 2010

MSC:

65K10

90C53

Keywords:

Unconstrained minimization

Symmetric rank-one update

Secant equation

Hessian approximation

ABSTRACT

Symmetric rank-one (SR1) is one of the competitive formulas among the quasi-Newton (QN) methods. In this paper, we propose some modified SR1 updates based on the modified secant equations, which use both gradient and function information. Furthermore, to avoid the loss of positive definiteness and zero denominators of the new SR1 updates, we apply a restart procedure to this update. Three new algorithms are given to improve the Hessian approximation with modified secant equations for the SR1 method. Numerical results show that the proposed algorithms are very encouraging and the advantage of the proposed algorithms over the standard SR1 and BFGS updates is clearly observed.

© 2010 Elsevier B.V. All rights reserved.

1. Introduction

We consider quasi-Newton (QN) methods for finding a local minimum of the unconstrained optimization problem

$$\min_{x \in \mathbb{R}^n} f(x), \quad (1.1)$$

where $f: \mathbb{R}^n \rightarrow \mathbb{R}$ is a smooth function. It will be assumed that f is continuous and at least twice differentiable. The basic framework of QN methods can be summarized as follows:

Given current iterate x_k , $\nabla f(x_k)$ or its finite difference approximation and $B_k \in \mathbb{R}^{n \times n}$ (secant approximation to $\nabla^2 f(x_k)$), we determine the QN direction p_k by

$$B_k p_k + \nabla f(x_k) = 0. \quad (1.2)$$

Once p_k is obtained, select new iterate x_{k+1} by a line search method based on the following equation

$$x_{k+1} = x_k + \alpha_k p_k. \quad (1.3)$$

Update B_k to B_{k+1} such that B_{k+1} is symmetric and satisfies the secant equation

$$B_{k+1} s_k = y_k, \quad (1.4)$$

where $s_k = x_{k+1} - x_k$ and $y_k = f(x_{k+1}) - f(x_k)$.

One of the popular ways to incorporate good curvature information of the objective function f into the updated matrix is to force the matrix to satisfy in the general secant equation (1.4). Some kind of variants of secant equations have been

* Corresponding author. Tel.: +60 173846343; fax: +60 389437958.

E-mail addresses: farzin@math.upm.edu.my, farzin.modarres@gmail.com (F. Modarres).

considered to get a better approximation matrix B_k . Yamaki and Yabe [1] proposed updating formula that satisfies the extended secant equation

$$B_{k+1}s_i = y_i, \quad i = 1, \dots, k,$$

which guarantees the n step termination property for minimizing a convex quadratic function [2]. Ford and Moghrabi [3] introduced multi-step method in which data from the m recent steps is employed in the construction of an interpolating path $x(\theta)$, in which the standard secant equation corresponds to $m = 1$. However, if we use not only the gradients but also the function values, we could have a better theoretical advantage than the general equation in the sense that their secant equation possesses a more accurate curvature information. For this purpose, Wei et al. [4] proposed the modified secant condition

$$B_{k+1}s_k = y_k^*, \quad (1.5)$$

where $y_k^* = y_k + A_k s_k$, which uses both gradient and function value information in order to get a higher order accuracy in approximating the second curvature of the objective function. Extensive numerical results have been reported that modified BFGS update based on the modified secant equation (1.5) gives improvement over the standard BFGS update.

In particular, Li and Fukushima [5,6] made some slight modifications to the standard BFGS method and developed a modified BFGS method and a cautious BFGS method to improve the global convergence property of the BFGS method. Under appropriate conditions, both methods are globally and superlinearly convergent for non-convex minimization problems. In fact, Li and Fukushima [5] have given a modified BFGS method which ensures the global convergence for a general objective function f by using the choice $A_k = (O\|\nabla f(x_k)\|)$. However, numerical results show that this choice does not outperform the BFGS method even we choose A_k to be a very small number.

In this paper, we consider the SR1 update for the Hessian approximation

$$B_{k+1} = B_k + \frac{(y_k - B_k s_k)(y_k - B_k s_k)^T}{(y_k - B_k s_k)^T s_k}, \quad (1.6)$$

and throughout, for a purpose of comparison, we also give the BFGS update

$$B_{k+1} = B_k - \frac{B_k s_k s_k^T B_k}{s_k^T B_k s_k} + \frac{y_k y_k^T}{y_k^T s_k}. \quad (1.7)$$

For the background on these updates, see [7–9]. The advantage of the SR1 method over others is in a reduction in the number of iterations. Such a reduction has been observed by many authors. Investigation in [10] on the SR1 update showed that when the SR1 and BFGS updates are available simultaneously, the SR1 update is more efficient. Meanwhile, the sequence of B_k can converge to the actual Hessian matrix at the solution $\nabla^2 f(x_*)$, provided that the steps taken are uniformly linearly independent, that the SR1 update denominator is always sufficiently different from zero, and that the iterates converge to a finite limit. Khalfan et al. [11] discussed their computational experience in using the SR1 update with the conventional line search and trust region algorithms. The results showed that in practice when the SR1 update solves a given problem, its efficiency is at least, if not better, as good as the BFGS update. As we know two basic disadvantages of the SR1 update are that the SR1 update may not preserve positive definiteness even when it is updated from a positive definite approximation. Furthermore, the SR1 update can be undefined. Many authors proposed modification to the SR1 update to preserve positive definiteness; see, for example, [12]. Successful numerical result test in a trust region framework to avoid the possible loss of positive definiteness have resulted in a renewed interest in SR1 update; see, for example, [13]. Phua [14] proposed some switching algorithms in which the SR1 update could be, when necessary, substitute with a rank-two Broyden class update. Another important approach of avoiding the loss of positive definiteness of the SR1 update is to scale/size the current update; see [15–17]. Osborne and Sun [17] discussed a rank-one type update of the form $B_{k+1} = \theta B_k + k_s w_k w_k^T$, where $w_k = y_k - \theta B_k s_k$, $k_s = 1/s_k^T w_k$, and θ is a scalar scaling parameter that can be chosen to ensure that the update is positive definite. Recently, Leong and Hassan [18] used a restarting procedure with a positive multiple of the identity matrix whenever these difficulties arise. Their results indicate that the restarting procedure with a simple initial scaling is more effective than restarting the update with the identity matrix in dealing with non-positive definite updates and is very competitive with the BFGS method. In fact, a considerable amount of works has been directed toward preserving the positive definiteness of SR1 update and not much is done in improving the quality of the Hessian approximation. Hence our motivation here is to modify the standard secant equation and incorporate the new modified secant equation to the SR1 update with a simple restart. We also study the numerical properties of the new methods.

This paper is organized as follows. In the next section, we construct the modified secant equation. In Section 2, we introduce the SR1 update obeying the modified secant equation. In Section 3, we present a scaling factor for the restarting iterations. In Section 4, we describe our algorithms for the modified SR1 update and the BFGS method. Then we investigate on the global convergence result in Section 5. Finally, in Section 6 the preliminary numerical results for three algorithms are given, and we compare these numerical results with the BFGS and SR1 methods.

2. Modified secant equations

In the following subsection, we first give the modified secant equation proposed in [4] and present our simple modification to the secant equation.

2.1. Construction of \tilde{y}_k

To generate a new secant update, we may generate a direction p_k from the QN direction (1.2) in which B_k is replaced by the matrix

$$\tilde{B}_k = B_k + \mu_k I, \quad (2.1)$$

where I is the unit matrix, and μ_k is chosen in a suitable way (practical choice of μ_k will be discussed in the next subsection). Since B_{k+1} satisfies

$$B_{k+1}(x_{k+1} - x_k) \approx \nabla f(x_{k+1}) - \nabla f(x_k),$$

the matrix \tilde{B}_{k+1} will satisfy the relation

$$\tilde{B}_{k+1}(x_{k+1} - x_k) = (B_{k+1} + \mu_k I)(x_{k+1} - x_k) \approx \nabla f(x_{k+1}) - \nabla f(x_k),$$

or equivalently

$$B_{k+1}s_k \approx y_k + \mu_k s_k.$$

So we can write the modified secant equation as follows

$$B_{k+1}s_k = y_k^*; \quad \text{where } y_k^* = y_k + \mu_k s_k. \quad (2.2)$$

It is obvious that if k approaches to infinity and μ_k to zero, then the modified secant equation is close to the general secant equation.

Now assume that the objective function f is smooth enough. Let $x_{k+1} = x_k + s_k$, by using the Taylor series along s_k we have

$$f(x_k) = f(x_{k+1}) - s_k^T \nabla f(x_{k+1}) + \frac{1}{2} s_k^T \nabla^2 f(x_{k+1}) s_k + O(\|s_k\|^3),$$

or

$$f(x_k) \cong f(x_{k+1}) - s_k^T \nabla f(x_{k+1}) + \frac{1}{2} s_k^T \nabla^2 f(x_{k+1}) s_k.$$

Therefore,

$$\begin{aligned} s_k^T \nabla^2 f(x_{k+1}) s_k &= 2(f(x_k) - f(x_{k+1})) + 2(\nabla f(x_{k+1}))^T s_k \\ &= 2(f(x_k) - f(x_{k+1})) + (\nabla f(x_{k+1}) + \nabla f(x_k))^T s_k + s_k^T y_k. \end{aligned} \quad (2.3)$$

By (2.2), we have

$$s_k^T B_{k+1} s_k = s_k^T y_k^* = s_k^T y_k + \mu_k s_k^T s_k, \quad (2.4)$$

and by comparing (2.3) with (2.4) we can choose μ_k in a reasonable way that it will satisfy in the following equation

$$\mu_k s_k^T s_k = \psi_k, \quad (2.5)$$

where $\psi_k = 2(f(x_k) - f(x_{k+1})) + (\nabla f(x_{k+1}) + \nabla f(x_k))^T s_k$.

From (2.5), we have a good choice of μ_k as follows

$$\mu_k s_k = w_k; \quad \text{where } w_k = \frac{\psi_k}{s_k^T s_k} u_k, \quad (2.6)$$

and $u_k \in \mathbb{R}^n$ is any vector such that $s_k^T u_k \neq 0$.

Wei et al. proposed the following equation based on the above relation

$$B_{k+1}s_k = y_k^*; \quad \text{where } y_k^* = y_k + \frac{\psi_k}{s_k^T u_k} u_k. \quad (2.7)$$

Throughout this paper the norm considered is the usual l_2 -norm on vectors and the corresponding induced norm on matrices.

The following theorem shows good properties of the secant equation (2.7) which can be found in [4].

Theorem 1. Assume that the function f is sufficiently smooth and μ_k satisfies (2.5). If $\|s_k\|$ is sufficiently small, then the following estimates hold

$$s_k^T (\nabla^2 f(x_{k+1}) s_k - y_k^*) = \frac{1}{3} s_k^T (T_{k+1} s_k) s_k + O(\|s_k\|^4), \quad (2.8)$$

$$s_k^T (\nabla^2 f(x_{k+1}) s_k - y_k) = \frac{1}{2} s_k^T (T_{k+1} s_k) s_k + O(\|s_k\|^4), \quad (2.9)$$

where $y_k^* = y_k + \mu_k s_k$, T_{k+1} is the tensor of f at x_{k+1} and

$$s_k^T (T_{k+1} s_k) s_k = \sum_{i,j,l=1}^n \frac{\partial^3 f(x_{k+1})}{\partial x^i \partial x^j \partial x^l} s_k^i s_k^j s_k^l.$$

Proof. By using the Taylor formula for the objective function f we get

$$f(x_k) = f(x_{k+1}) - (\nabla f(x_{k+1}))^T s_k + \frac{1}{2} s_k^T \nabla^2 f(x_{k+1}) s_k - \frac{1}{6} s_k^T (T_{k+1} s_k) s_k + O(\|s_k\|^4),$$

and

$$(\nabla f(x_k))^T s_k = (\nabla f(x_{k+1}))^T s_k - s_k^T \nabla^2 f(x_{k+1}) s_k + \frac{1}{2} s_k^T (T_{k+1} s_k) s_k + O(\|s_k\|^4).$$

Then the conclusion follows by the definitions of y_k and y_k^* . \square

The above theorem shows that y_k^* is a better approximation to $\nabla^2 f(x_{k+1}) s_k$ than y_k .

2.2. A simple modification to the secant equation (2.7)

Since $s_k^T y_k^* = s_k^T y_k + \mu_k$ and μ_k is possibly negative, if $\mu_k < 0$ is larger than $s_k^T y_k$, then $s_k^T y_k^* < 0$. To cope with this defect, we further modify the vector y_k^* , as

$$\tilde{y}_k = y_k + \text{sgn}(\psi_k) \frac{\psi_k}{s_k^T u_k} u_k. \quad (2.10)$$

Thus it is easy to see that we always have $s_k^T y_k^* > 0$ if and only if $s_k^T y_k > 0$.

Here we will give our three choices of vector u_k in (2.10) for the modified secant equations while Wei et al. considered just two cases in (2.7).

(i) Putting $u_k = s_k$ in (2.10) we have

$$\tilde{y}_k = y_k + \text{sgn}(\psi_k) \frac{\psi_k}{s_k^T s_k} s_k. \quad (2.11)$$

(ii) Putting $u_k = y_k$ in (2.10) we have

$$\tilde{y}_k = y_k + \text{sgn}(\psi_k) \frac{\psi_k}{s_k^T y_k} y_k. \quad (2.12)$$

(iii) Putting $u_k = \nabla f(x_k)$ in (2.10) we have

$$\tilde{y}_k = y_k + \text{sgn}(\psi_k) \frac{\psi_k}{s_k^T \nabla f(x_k)} \nabla f(x_k). \quad (2.13)$$

3. The SR1 update with modified secant equation

In this section, we consider the modified SR1 update with the modified secant equations (2.10). Furthermore, we will introduce a strategy for preserving the positive definiteness in the modified SR1 formula.

3.1. Modified SR1 update

Now we propose the modified SR1 update based on modified secant equations (2.10),

$$B_{k+1} = B_k + \frac{(\tilde{y}_k - B_k s_k)(\tilde{y}_k - B_k s_k)^T}{(\tilde{y}_k - B_k s_k)^T s_k}, \quad (3.1)$$

where $\tilde{y}_k = y_k + \text{sgn}(\psi_k) \frac{\psi_k}{s_k^T u_k} u_k$. If we approximate the inverse of the Hessian by H_k , the modified secant equation is rewritten as

$$s_k = H_{k+1} \tilde{y}_k. \quad (3.2)$$

Based on the equation, we have the inverse SR1 update

$$H_{k+1} = H_k + \frac{(s_k - H_k \tilde{y}_k)(s_k - H_k \tilde{y}_k)^T}{(s_k - H_k \tilde{y}_k)^T \tilde{y}_k}. \quad (3.3)$$

We call the updates (3.1) as the modified SR1 update and (3.3) is its inverse version.

3.2. Positive definiteness of the modified SR1 update

In this subsection, we will apply a restart procedure to the symmetric rank-one update to avoid the loss of positive definiteness. The important difference between the restarting approach and the commonly used sizing approach by [15] is that sizing is incorporated to each SR1 updates in preserving positive definiteness while the restarting approach restarts the SR1 algorithm with a multiple of the identity matrix whenever SR1 updates are not positive definite. The main reason that we do not prefer sizing is that, it requires the computation of B_k (the approximation of Hessian). If the SR1 matrix in that iteration is positive definite, sizing will be redundant.

To cope with this defect, we employ \tilde{y} to the optimal scaling factor of Leong and Hassan [18] and consider the scaled identity matrix defined by $\tilde{\lambda}_{k-1}I$ where

$$\tilde{\lambda}_{k-1} = \frac{s_{k-1}^T s_{k-1}}{\tilde{y}_{k-1}^T s_{k-1}} - \left\{ \frac{(s_{k-1}^T s_{k-1})^2}{(\tilde{y}_{k-1}^T s_{k-1})^2} - \frac{s_{k-1}^T s_{k-1}}{\tilde{y}_{k-1}^T \tilde{y}_{k-1}} \right\}^{1/2}. \quad (3.4)$$

Our method is to incorporate restart procedure, whenever we loss positive definiteness in the modified SR1 or the denominator in the modified SR1 is nearly zero or $\{H_k\}$ is unbounded. Now with this modification, we present our new algorithms:

4. Description of algorithms

MSR1(I): Modified SR1 Algorithm with the choice $u_k = s_k$

Step 0. Given an initial point x_0 , an initial positive definite matrix $H_0 = I$, set $k = 0$.

Step 1. If the convergence criterion $\|\nabla f(x_k)\| \leq \varepsilon \times \max(1, \|x_k\|)$ is achieved, then stop.

Step 2. Compute a quasi-Newton direction by $p_k = -H_k \nabla f(x_k)$.

Step 3. If

$$s_k^T \tilde{y}_k - \tilde{y}_k^T H_k \tilde{y}_k < 0; \quad (H_k \text{ might not be positive definite}), \quad (4.1)$$

or

$$|\tilde{y}_k^T (s_k - H_k \tilde{y}_k)| < r \|\tilde{y}_k\| \|s_k - H_k \tilde{y}_k\|, \quad (4.2)$$

where $r \in (0, 1)$, (denominator in H_k is sufficiently close to zero; see [19]) or

$$\|H_k\|_\infty > L; \quad (\text{where } L \text{ is a preset constant}), \quad (4.3)$$

set $H_k = \tilde{\lambda}_{k-1}I$, where $\tilde{\lambda}_{k-1}$ is given in (3.4).

Step 4. Find an acceptable steplength such that the Wolfe conditions

$$f(x_k + \alpha_k p_k) \leq f(x_k) + \delta_1 \alpha_k \nabla f(x_k)^T p_k, \quad (4.4)$$

$$\nabla f(x_k + \alpha_k p_k)^T p_k \geq \delta_2 \nabla f(x_k)^T p_k, \quad (4.5)$$

where $0 < \delta_1 < \delta_2 < 1$, $\delta_1 < \frac{1}{2}$, are satisfied.

($\alpha_k = 1$ is always tried first, $\delta_1 = 10^{-4}$, $\delta_2 = 0.9$).

Step 5. Set $x_{k+1} = x_k + \alpha_k p_k$.

Step 6. Calculate \tilde{y}_k by using the following equation

$$\tilde{y}_k = y_k + \text{sgn}(\psi_k) \frac{\psi_k}{s_k^T s_k} s_k,$$

where $\psi_k = 2(f(x_k) - f(x_{k+1})) + (\nabla f(x_{k+1}) + \nabla f(x_k))^T s_k$.

Step 7. Compute the next inverse Hessian approximation H_{k+1} by (3.3).

Step 8. Set $k = k + 1$, and go to Step 1.

MSR1(II): Modified SR1 Algorithm with the choice $u_k = y_k$.

Step 6 in MSR1(I) is replaced by: Calculate \tilde{y}_k by using the following equation

$$\tilde{y}_k = y_k + \text{sgn}(\psi_k) \frac{\psi_k}{s_k^T y_k} y_k, \quad (4.6)$$

where

$$\psi_k = 2(f(x_k) - f(x_{k+1})) + (\nabla f(x_{k+1}) + \nabla f(x_k))^T s_k. \quad (4.7)$$

MSR1(III): Modified SR1 Algorithm with the choice $u_k = \nabla f(x_k)$.

Step 6 in MSR1(I) is replaced by: Calculate \tilde{y}_k by using the following equation

$$\tilde{y}_k = y_k + \text{sgn}(\psi_k) \frac{\psi_k}{s_k^T \nabla f(x_k)} \nabla f(x_k), \quad (4.8)$$

where

$$\psi_k = 2(f(x_k) - f(x_{k+1})) + (\nabla f(x_{k+1}) + \nabla f(x_k))^T s_k. \quad (4.9)$$

5. Global convergence result

Throughout this section, the following assumptions will be considered about the objective function f .

Assumption 1. Let G be the matrix of second derivatives of f .

- The objective function f is continuously differentiable in a neighborhood F of the level set $D = \{x \in \mathbb{R}^n : f(x) \leq f(x_0)\}$ and bounded below in \mathbb{R}^n .
- The gradient is Lipschitz continuous, i.e., there exists a constant $L > 0$ such that

$$\|\nabla f(x_2) - \nabla f(x_1)\| \leq L\|x_2 - x_1\|, \quad (5.1)$$

for all $x_1, x_2 \in F$.

Let us define

$$\cos(\theta_k) = -\frac{\nabla f(x_k)^T p_k}{\|\nabla f(x_k)\| \|p_k\|}. \quad (5.2)$$

Theorem 2. Let x_0 be a starting point for which f satisfies Assumption 1. Consider $\{x_k\}$ the sequence of points generated by the updating scheme $x_{k+1} = x_k + \alpha_k p_k$ where the sequence $\{B_k\}$ is generated by MSR1(I)–MSR1(III) and α_k satisfies the Wolfe conditions (4.4)–(4.5). Then

$$\sum_{k=1}^{\infty} \cos^2 \theta_k \|\nabla f(x_k)\|^2 < \infty. \quad (5.3)$$

Proof. From (4.5) we have that

$$(\nabla f(x_{k+1}) - \nabla f(x_k))^T p_k \geq (\delta_2 - 1)^T \nabla f(x_k)^T p_k.$$

Moreover, from Lipschitz condition (5.1) we obtain

$$(\nabla f(x_{k+1}) - \nabla f(x_k))^T p_k \leq \alpha_k L \|p_k\|^2.$$

Combining the above inequalities gives

$$\alpha_k \geq \left(\frac{\delta_2 - 1}{L} \right) \nabla f(x_k)^T p_k / \|p_k\|^2. \quad (5.4)$$

Therefore, by using the first Wolfe condition (4.4) and (5.4), we have

$$f_{k+1} \leq f_k + \delta_1 \left(\frac{\delta_2 - 1}{L} \right) (\nabla f(x_k)^T p_k)^2 / \|p_k\|^2.$$

We now use definition (5.2) to write this relation as

$$f_{k+1} \leq f_k + c \cos^2 \theta_k \|\nabla f(x_k)\|^2,$$

where $c = \frac{\delta_1(\delta_2-1)}{L}$. Summing this expression and recalling that f is bounded below we obtain

$$\sum_{k=1}^{\infty} \cos^2 \theta_k \|\nabla f(x_k)\|^2 < \infty,$$

which concludes the proof. \square

Since Step 3 in MSR1(I)–MSR1(III) maintains p_k as a descent direction and the matrices B_k are uniformly bounded, i.e. that for all k

$$\|B_k\| \|B_k^{-1}\| \leq \Delta$$

for some constant $\Delta > 0$. Then from (5.2) we have that

$$\cos \theta_k \geq 1/\Delta.$$

Thus we conclude directly from (5.3) that

$$\lim_{k \rightarrow \infty} \|\nabla f(x_k)\| = 0.$$

6. Numerical result

In this section, we give the numerical result using the above outlined algorithms. We tested the MSR1(I)–MSR1(III) on a variety of 49 test problems selected from [19,20] with dimensions, $2 \leq n \leq 1000$.

A total of 268 runs were performed. All the algorithms are implemented in Fortran 77. In all cases, double precision arithmetic were used. The parameters in Wolfe conditions are set as $\delta_1 = 10^{-4}$ and $\delta_2 = 0.9$ where $\alpha_k = 1$ is always tried first. The stopping tolerance, ε used was 10^{-5} . We also force the routine to stop if the number of function evaluations exceed 1000.

As we know the BFGS and SR1 methods are considered to be efficient methods among the QN methods, and thus we compare our algorithms with the SR1 and BFGS methods. Restarting strategy of [18] is also implied within the standard SR1 method. In this series of experiments, MSR1(I)–MSR1(III) can solve over 96% of the test problems, BFGS solves 94% and SR1 solves 84%.

Since we use a large set of test problems, we choose to summarize our results by using the Geometric and Arithmetic means of the number of iterations and function/gradient evaluations required to solve these problems by MSR1(I)–MSR1(III) to the corresponding mean for the BFGS and SR1 methods. In Tables 1 and 2, “Itrn”, “Feval”, and “CPU” means iterations, function /gradient calls and CPU time, respectively.

Table 1

Ratio of MSR1(I)–MSR1(III) Cost to BFGS Cost.

| Mean | MSR1(I) | | | MSR1(II) | | | MSR1(III) | | |
|------------|---------|--------|------|----------|--------|------|-----------|--------|------|
| | Itrn. | Feval. | CPU | Itrn. | Feval. | CPU | Itrn. | Feval. | CPU |
| Arithmetic | 0.82 | 0.88 | 0.97 | 0.87 | 0.94 | 0.96 | 0.86 | 0.91 | 0.97 |
| Geometric | 0.72 | 0.82 | 0.95 | 0.82 | 0.87 | 0.93 | 0.81 | 0.84 | 0.94 |

Table 2

Ratio of MSR1(I)–MSR1(III) Cost to SR1 Cost.

| Mean | MSR1(I) | | | MSR1(II) | | | MSR1(III) | | |
|------------|---------|--------|------|----------|--------|------|-----------|--------|------|
| | Itrn. | Feval. | CPU | Itrn. | Feval. | CPU | Itrn. | Feval. | CPU |
| Arithmetic | 0.78 | 0.85 | 0.88 | 0.83 | 0.87 | 0.87 | 0.81 | 0.88 | 0.87 |
| Geometric | 0.71 | 0.78 | 0.85 | 0.79 | 0.80 | 0.85 | 0.78 | 0.80 | 0.84 |

Tables 1 and 2 show that MSR1(I)–MSR1(III) improve significantly over the performance of their original SR1 counterparts. On the other hand, the improvements of MSR1(I)–MSR1(III) over BFGS are 13, 6 and 3%, in average, respectively in terms of the number of iterations, function/gradient calls and CPU time. Similarly, the improvements of MSR1(I)–MSR1(III) over SR1 are 17, 12 and 13%, in average, respectively, in terms of the number of iterations and function/gradient calls.

Therefore, MSR1(I)–MSR1(II) are 6%–17% in average, faster and cheaper than the BFGS and SR1 methods.

Comparing the performance of all these algorithms, Tables 1 and 2 show that MSR1(I) scores the best while MSR1(III) is the second best, and the MSR1(II) is the third best, with SR1 the last and BFGS the second last.

Also in order to compare and evaluate the performance of our algorithms with that of SR1 and BFGS, we use the performance profiling proposed in [21].

In Figs. 1–3, we plot the performance profiles for the algorithms in terms of the number of iterations, function and gradient evaluations and CPU time, respectively.

From the figures, we observe that the performance of MSR1(I) is the best among the other methods.

Also they show that MSR1(II) and MSR1(III) are very competitive with BFGS method and vast superior than the SR1 method. It is not surprising that the proposed methods can be faster than the SR1 because a higher order accuracy in approximating the Hessian matrix of the objective function makes MSR1(I)–MSR1(III) need fewer iterations, less function and gradient evaluations. Fig. 3 indicates that relative to CPU time metric, the proposed methods are fastest, followed by the BFGS and the SR1.

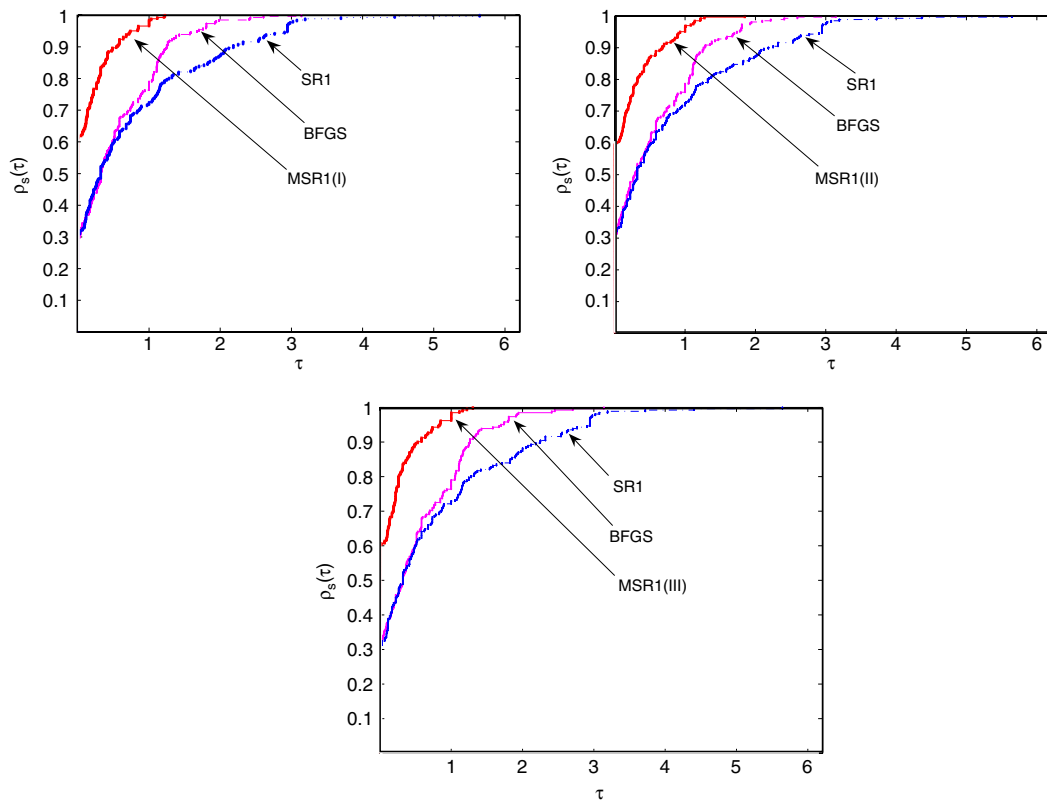


Fig. 1. Performance profile based on iterations.

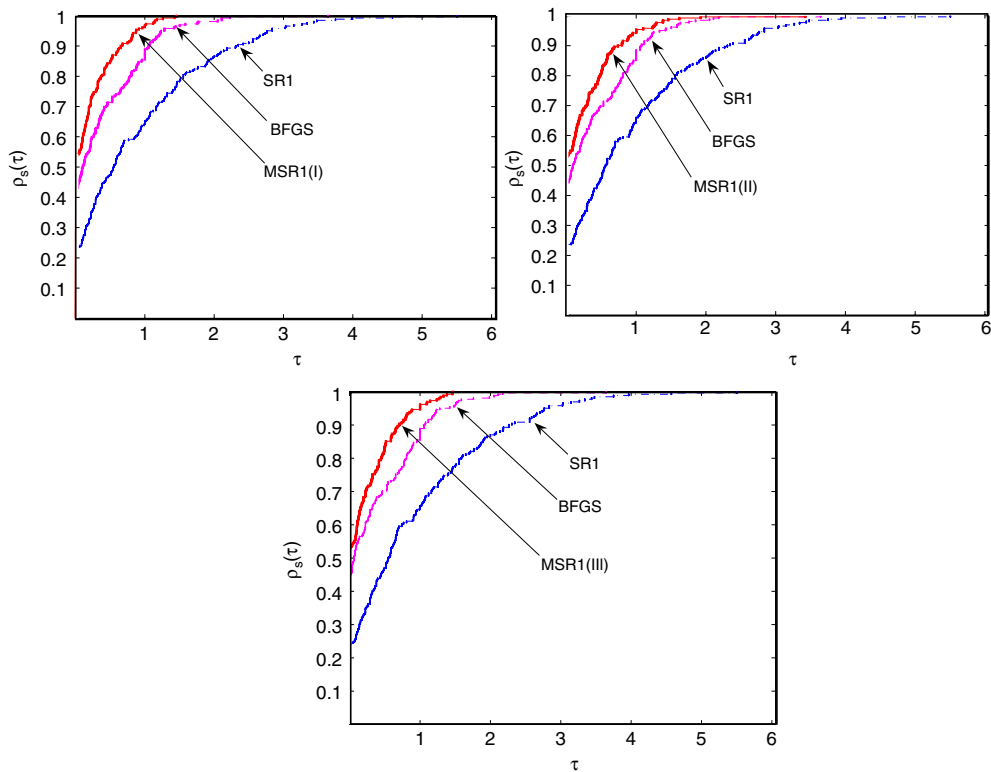


Fig. 2. Performance profile based on function/gradient evaluations.

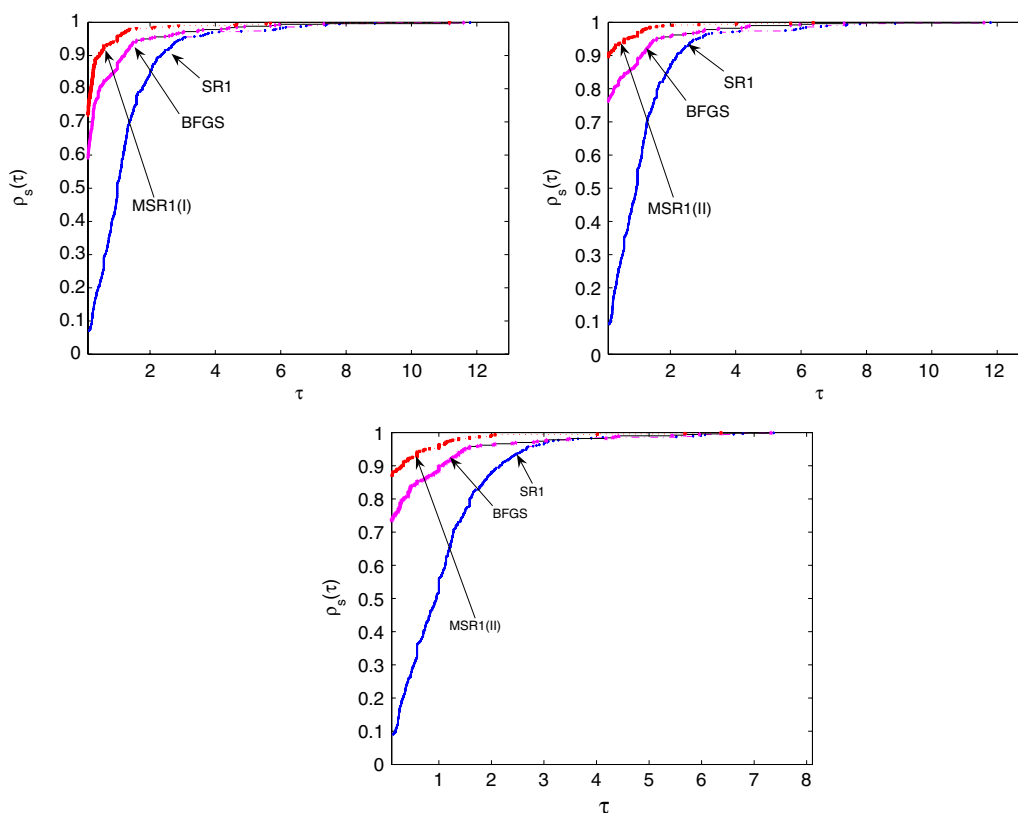


Fig. 3. Performance profile based on CPU time.

7. Conclusion

We have presented three modified SR1 updates based on modified secant equations and considered a restart procedure to preserve positive definiteness and to avoid zero denominators. Compared with the SR1 and BFGS methods, our algorithms require fewer iteration and function/gradient calls. Moreover, most of the problems can be solved by MSR1(I)–MSR1(III) within a given number of iterations and function/gradient calls but not for the SR1 method. Therefore, we conclude that by improving the Hessian approximation of the SR1 methods, the performance of the modified SR1 method can be enhanced dramatically.

References

- [1] N. Yamaki, H. Yabe, A family of the quasi-Newton methods, *TRU Math.* (16) (1980) 49–54.
- [2] H.Y. Huang, Unified approach to quadratically convergent algorithms for function minimization, *J. Optim. Theory Appl.* (5) (1970) 405–423.
- [3] J.A. Ford, I.A. Moghrabi, Multi-step quasi-Newton methods for optimization, *J. Comput. Appl. Math.* 50 (1994) 305–323.
- [4] Z. Wei, G. Li, L. Qi, New quasi-Newton methods for unconstrained optimization problems, *J. Appl. Math. Comput. Theory Appl.* 175 (2006) 1156–1188.
- [5] D. Li, M. Fukushima, A modified BFGS method and its global convergence in nonconvex minimization, *J. Comput. Appl. Math.* 129 (2001) 15–35.
- [6] D. Li, M. Fukushima, On the global convergence of the BFGS method for nonconvex unconstrained optimization problems, *SIAM J. Optim.* 11 (2001) 1054–1064.
- [7] J.E. Dennis, R.B. Schnabel, *Numerical Methods for Nonlinear Equations and Unconstrained Optimization*, Prentice Hall, Englewood Cliffs, 1982.
- [8] R. Fletcher, *Practical Methods of Optimization*, Wiley, New York, 1980.
- [9] P.E. Gill, W. Murray, M.H. Wright, *Practical Optimization*, Academic, London, 1981.
- [10] A.R. Conn, N.I.M. Gould, Ph.L. Toint, Convergence of quasi-Newton matrices generated by the symmetric rank one update, *Math. Program.* 48 (1991) 549–560.
- [11] H. Khalfan, R.H. Byrd, R.B. Schnabel, A theoretical and experimental study of the symmetric rank one update, *SIAM J. Optim.* 3 (1993) 1–24.
- [12] P. Spellucci, A modified rank one update which converges Q -superlinearly, *Comput. Optim. Appl.* 19 (2001) 273–296.
- [13] C.M. Ip, M.J. Todd, Optimal conditioning and convergence in rank one quasi-Newton updates, *SIAM J. Numer. Anal.* 25 (1988) 206–221.
- [14] P.K.H. Phua, Eigenvalues and switching algorithms for quasi-Newton updates, *Optimization* 42 (1997) 185–217.
- [15] J.E. Dennis, H. Wolkowicz, Sizing and least change secant methods, *SIAM J. Numer. Anal.* 30 (5) (1993) 1291–1313.
- [16] H.F.H. Khalfan, *Topics in quasi-Newton methods for unconstrained optimization*, Ph.D. Thesis, University of Colorado, Colorado, 1989.
- [17] M.R. Osborne, L. Sun, A new approach to symmetric rank-one updating, *IMA J. Numer. Anal.* 19 (1999) 497–507.
- [18] W.J. Leong, M.A. Hassan, A restarting approach for the symmetric rank one update for unconstrained optimization, *Comput. Optim. Appl.* 42 (3) (2009) 327–334.
- [19] N. Andrei, An unconstrained optimization test functions collection, *Adv. Model. Optim.* 10 (2008) 147–161.
- [20] I. Bongartz, A.R. Conn, N.I.M. Gould, Ph.L. Toint, CUTE: constrained and unconstrained testing environment, *ACM Trans. Math. Software* 21 (1995) 123–160.
- [21] E.D. Dolan, J.J. Moré, Benchmarking optimization software with performance profiles, *Math. Program.* 91 (2) (2002) 201–213.