Journal of Mathematical Modeling
Vol. 11, No. 2, 2023, pp. 323–342. Research Article

JMM

A new version of augmented self-scaling BFGS method

Mohamad Jourak[†], **Saeed Nezhadhosein**[†*], **Farzad Rahpeymaii**[‡]

[†]*Department of Mathematics, Payame Noor University, P.O. Box. 19395-3697, Tehran, Iran*
[‡]*Department of Mathematics, Technical and Vocational University (TVU), Tehran, Iran*
Email(s): mjourak@student.pnu.ac.ir, s_nejhadhosein@pnu.ac.ir, rahpeyma_83@yahoo.com

Abstract. A new version of the augmented self-scaling memoryless BFGS quasi-Newton update, proposed in [Appl. Numer. Math. 167, 187–201, (2021)], is suggested for unconstrained optimization problems. To use the corresponding scaled parameter, the clustering of the eigenvalues of the approximate Hessian matrix about one point is applied with three approaches. The first and second approaches are based on the trace and the determinant of the matrix. The third approach is based on minimizing the measure function. The sufficient descent property is guaranteed for uniformly convex functions, and the global convergence of the proposed algorithm is proved both for the uniformly convex and general nonlinear objective functions, separately. Numerical experiments on a set of test functions of the CUTEr collection show that the proposed method is robust. In addition, the proposed algorithm is effectively applied to the salt and pepper noise elimination problem.

Keywords: Unconstrained optimization, augmented BFGS, noise elimination problem.
AMS Subject Classification 2010: 90C34, 90C40

1 Introduction and motivation

Let $f : \mathbb{R}^n \to \mathbb{R}$ be a continuously differentiable function. Then consider the unconstrained optimization (UO) problem $\min_{x\in\mathbb{R}^n} f(x)$. Many applicable problems of science and engineering can be formulated as UO models, such as electromagnetic energy [22], neural networks [24], image processing [12], and signal processing [29] (for more cases, see [1, 33]).

Analytical methods are poor in solving UO problems, especially in high-dimension and for extremely nonlinear objective functions. Therefore, iterative numerical methods are famous in this context. Two main families of iterative algorithms in this field are line search (LS) and trust region (TR) methods.

*Corresponding author

Received: 10 December 2022 / Revised: 27 March 2023 / Accepted: 03 April 2023
DOI: 10.22124/jmm.2023.23425.2089

Here, we use an LS method with the following iterative formula

$$x_{k+1} = x_k + \alpha_k d_k, \qquad k \geq 0, \tag{1}$$

where α_k is a step length and d_k is a descent quasi-Newton (QN) direction, which is the solution of the following system of linear equations

$$B_k d_k = -g_k, \qquad k \geq 0, \tag{2}$$

where $g_k = \nabla f(x_k)$ and B_k is the positive-definite symmetric approximate of the Hessian matrix. Usually, the step length α_k in (1) is determined by Wolfe LS conditions

$$f(x_k + \alpha_k d_k) \leq f(x_k) + \delta \alpha_k g_k^T d_k, \tag{3}$$

$$\nabla f(x_k + \alpha_k d_k)^T d_k \geq \sigma g_k^T d_k, \tag{4}$$

with $0 < \delta < \sigma < 1$.

For QN methods, the sequence of Hessian approximations $\{B_k\}_{k\geq 0}$, starting from an initial positive-definite matrix $B_0 \in \mathbb{R}^{n\times n}$ is updated to satisfy the following so-called secant condition

$$B_{k+1} s_k = y_k, \qquad \text{for all } \ k \geq 0, \tag{5}$$

where $s_k = x_{k+1} - x_k$ and $y_k = g_{k+1} - g_k$. In addition to the secant condition in (5), some QN updates must preserve positive definiteness. Moreover, theoretical and numerical aspects of QN methods are discussed in [18, 19].

As we know, QN updates, based on the previous approximation of the Hessian, are used in two main approaches, containing full and diagonal matrices [4]. In the full case, the general update formula is devoted to the Broyden family [13]

$$B_{k+1} = B_k - \frac{B_k s_k s_k^T B_k}{s_k^T B_k s_k} + \frac{y_k y_k^T}{s_k^T y_k} + \beta u_k u_k^T, \tag{6}$$

where β is the measure parameter and

$$u_k = (s_k^T B_k s_k)^{1/2} \left(\frac{y_k}{s_k^T y_k} - \frac{B_k s_k}{s_k^T B_k s_k} \right).$$

In the update (6), for $\beta = 0$, $\beta = 1$, and $\beta = 1/(1 - \frac{s_k^T B_k s_k}{s_k^T y_k})$, the method is reduced to Broyden–Felether–Goldfarb–Shanno (BFGS), Davidon–Felether–Powell (DFP), and symmetric rank one (SR1), respectively [37]. Among them, the BFGS is the most efficient method for solving medium-dimension UO problems [14]. The global convergence of the method has been proved for convex functions [15, 16]. Despite very important and valuable features of the self-correction property of the BFGS method, this method may not be converged for general functions. To overcome this defect, some researchers introduced modified versions of BFGS (see [2] for more details). For instance, using a modified secant equation (MSE) instead of the standard version of this equation is common in literatures [33]. This idea is used to improve conjugate gradient (CG) methods [31] and spectral CG methods [32]. For the BFGS update, MSE can be used for more accurate approximation of inverse Hessian, incorporating the function and the gradient in the secant equation (5). Recently, as an extension of MSEs proposed by Wei et al. [39], Zhang et al. [43], and Li and Fukushima [25] and Arazm et al. [7] introduced an extended MSE by the following equation

$$B_{k+1}s_k = \tilde{y}_k, \tag{7}$$

where

$$\tilde{y}_k = y_k + \tau_k s_k, \quad \tau_k = \tau \frac{\bar{\theta}_k}{\|s_k\|^2} + C\|g_k\|^p, \quad \theta_k = 2(f_k - f_{k+1}) + s_k^T(g_k + g_{k+1}), \tag{8}$$

in which τ, C, and p are nonnegative constants and $\bar{\theta}_k = \max\{\theta_k, 0\}$. In (8), if $\tau = C = 0$, then the MSE is reduced to the standard secant equation. If $\tau = 0$ and $C > 0$, then (7) is reduced to the MSE proposed by Li and Fukushima [25]. Finally, if $C = 0$, then the choices $\tau = 3$ and $\tau = 1$ coincide to MSEs proposed by Zhang et al. [43] and Wei et al. [39], respectively. Resently, using MSE in (7), Babaie-Kafaki et al. [11] have introduced a QN method, based on the SR1 update. By the similar manner, Aminifard et al. [2] proposed an augmented BFGS, called the ABFGS update

$$B_{k+1}^{ABFGS} = B_{k+1}^{BFGS} + \tau_k \frac{s_k s_k^T}{s_k^T y_k}, \tag{9}$$

where τ_k is defined in (8). Moreover, using the Shermen-Morrison formula [37], we have

$$H_{k+1}^{ABFGS} = H_{k+1}^{BFGS} - \frac{\tau_k}{\gamma_k} \frac{z_k z_k^T}{s_k^T y_k}, \tag{10}$$

where $H_k = B_k^{-1}$ is the approximation of the inverse by the Hessian matrix at x_k and

$$z_k = -v_k y_k + \left(1 + v_k \frac{\|y_k\|^2}{s_k^T y_k}\right) s_k, \quad \gamma_k = \tau_k + \frac{s_k^T y_k}{\|s_k\|^2} + \tau_k v_k \left(\frac{\|y_k\|^2}{s_k^T y_k} - \frac{s_k^T y_k}{\|s_k\|^2}\right),$$

where $v_k > 0$ is a scaled parameter. It is notable that the ABFGS update in (9) can be considered as a rank-one modification of the BFGS update formula [2]. The self-scaling memoryless version of ABFGS, called SMABFGS, is obtained by setting $B_k = \frac{1}{v_k} I_n$ in (9) or $H_k = v_k I_n$ in (10), where I_n is the identity matrix. The update of the Hessian approximation and its inverse of SMABFGS are, respectively,

$$B_{k+1}^{SMABFGS} = \frac{1}{v_k} I_n - \frac{1}{v_k} \frac{s_k s_k^T}{s_k^T s_k} + \frac{y_k y_k^T}{s_k^T y_k} + \tau_k \frac{s_k s_k^T}{s_k^T s_k}, \tag{11}$$

$$H_{k+1}^{SMABFGS} = v_k I_n - v_k \frac{s_k y_k^T + y_k s_k^T}{s_k^T y_k} + \left(1 + v_k \frac{y_k^T y_k}{s_k^T y_k}\right) \frac{s_k s_k^T}{s_k^T y_k} - \frac{\tau_k}{\gamma_k} \frac{z_k z_k^T}{s_k^T y_k}. \tag{12}$$

It should be mentioned that the ABFGS formula (12) preserves the positive definiteness condition provided that H_k is positive-definite satisfying (7) [2]. However, similar to setting the spectral parameter of SMBFGS, selecting an appropriate parameter in SMABFGS is critical theoretically and numerically.

For the SMBFGS updates, the two well-known choices for scale parameters are chosen by Oren and Spedicator [35] and Oren and Luenberger [34], which are based on a two-point approximation of the standard secant equation (5). These parameters are as follows

$$v_k^{OS} = \frac{s_k^T y_k}{\|y_k\|^2}, \quad v_k^{OL} = \frac{\|s_k\|^2}{s_k^T y_k}. \tag{13}$$

Moreover, there are some ideas in literature for tuning this parameter, such as clustering the eigenvalues in the search direction matrix [2,3,5,40], minimizing the condition number [9,10], the measure function [5,20,38], the distance between the direction matrices [17,30], and the difference between the largest and the smallest eigenvalues [9,26]. Recently, Andrei [5] has introduced three procedures for determining the scale parameter in SMBFGS based on clustering the eigenvalues, using the determinate and the trace of the search direction matrix and minimizing the measure function of Byrd and Nocedal [15] (see [6,28]).

As we know, the scaled parameter of the SMABFGS has not been discussed in the previous studies. Hence, motivated by setting this parameter for the augmented version of the QN method, the SMBFGS updates in (11) and (12), a novel algorithm in the LS category for UO problems is introduced. The idea is to cluster the eigenvalues of the search direction matrix using three approaches, which contain evaluating the trace, the determinate, and the measure function of this matrix. The main features of the new algorithm are that the method enforces the sufficient descent property for uniformly convex functions and the global convergence for general functions.

The paper is organized as follows. In Section 2, using clustering of the eigenvalues of SMABFGS, a new scaling parameter is suggested. In Section 3, the convergence analysis of the new method is proved. In Section 4, two numerical experiments are made to demonstrate the efficiency of our algorithm. Finally, conclusions are given in Section 5.

2 A new version of the SMABFGS update

In this section, similar to [5], we use a one-point clustering approach for the SMABFGS method to set the scale parameter. In continuation, for simplification, we omit the up script notation of SMABFGS for both matrix and search direction. So, by rewriting (11) and (12), the approximations of Hessian and its inverse are, respectively, defined

$$B_{k+1} = \frac{1}{v_k} I_n + \frac{y_k y_k^T}{s_k^T y_k} + \left(\frac{\tau_k}{\|s_k\|^2} - \frac{1}{v_k \|s_k\|^2} \right) s_k s_k^T, \tag{14}$$

$$H_{k+1} = v_k I_n - \frac{v_k}{s_k^T y_k} \left(s_k y_k^T + y_k s_k^T \right) + \left(1 + \frac{v_k \|y_k\|^2}{s_k^T y_k} \right) \frac{s_k s_k^T}{s_k^T y_k} - \frac{\tau_k}{\gamma_k} \frac{z_k z_k^T}{s_k^T y_k}. \tag{15}$$

Now, by applying (15) to (2), the search direction

$$d_k = -v_k g_k + v_k \frac{s_k y_k^T + y_k s_k^T}{s_k^T y_k} g_k - \left(1 + v_k \frac{y_k^T y_k}{s_k^T y_k} \right) \frac{s_k s_k^T}{s_k^T y_k} g_k + \frac{\tau_k}{\gamma_k} \frac{z_k z_k^T}{s_k^T y_k} g_k, \tag{16}$$

is obtained, where τ_k, γ_k, and z_k are from (7) and (10). It is notable that for $\tau_k = 0$, the corresponding QN method is reduced to SMBFGS. As we know, the self-scaling memoryless version of the QN method is very sensitive on the scaled parameter, v_k. Therefore, using clustering of eigenvalues, similar to [5] for SMBFGS, we apply three approaches to choose this parameter in (14) or (15). First of all, we need to compute the determinant and the trace of the matrix B_{k+1} in (14). After some simple algebraic manipulations on (14), the trace can be obtained by

$$tr(B_{k+1}) = \frac{n-1}{v_k} + \frac{\|y_k\|^2}{s_k^T y_k} + \tau_k. \tag{17}$$

Moreover, using the determinate of the rank-two update, [37, Eq. (1.2.70)], the determinate of B_{k+1} in (14)

$$\det(B_{k+1}) = \frac{1}{v_k^n}\left[v_k^2\tau_k\left(\frac{\|y_k\|^2}{s_k^T y_k} - \frac{s_k^T y_k}{\|s_k\|^2}\right) + v_k\left(\tau_k + \frac{s_k^T y_k}{\|s_k\|^2}\right)\right], \tag{18}$$

is achieved. By applying an eigenvalue analysis, similar to [5], it can be deduced that the matrix B_{k+1} in (14) has the same $(n-2)$ eigenvalues equal to $\frac{1}{v_k}$, with $\rho_k^{\pm}$ as two remainder of them. Now, to obtain the scaled parameter, we adjust $\rho_k^{\pm}$ in a way that all eigenvalues are clustered to one point, that is $\rho_k^+ = \rho_k^- = \frac{1}{v_k}$. To impose this condition, in the first approach, we apply the trace of B_{k+1} to (17). Therefore, it uses the condition

$$tr(B_{k+1}) = \frac{n}{v_k}, \tag{19}$$

where $tr(B_{k+1})$ is given by (17) and we get

$$\frac{n-1}{v_k} + \frac{\|y_k\|^2}{s_k^T y_k} + \tau_k = \frac{n}{v_k}, \tag{20}$$

which leads to a new scaled parameter

$$v_k^{TR} = \frac{s_k^T y_k}{\|y_k\|^2 + \tau_k s_k^T y_k}. \tag{21}$$

The well definedness of v_k^{TR} in (21) can be guaranteed by the Wolfe LS conditions, given in (3)–(4), that guarantee $s_k^T y_k > 0$.

In the second approach, we use the determinant of B_{k+1} in (18). By applying similar manner of the trace approach, the condition

$$\det(B_{k+1}) = \frac{1}{v_k^n}, \tag{22}$$

can be imposed, which leads to the algebraic equation

$$\frac{1}{v_k^n}\left[v_k^2\tau_k\left(\frac{\|y_k\|^2}{s_k^T y_k} - \frac{s_k^T y_k}{\|s_k\|^2}\right) + v_k\left(\tau_k + \frac{s_k^T y_k}{\|s_k\|^2}\right)\right] = \frac{1}{v_k^n}.$$

By elimination v_k^n from both sides of this equation, we have

$$\tau_k\left(\frac{\|y_k\|^2}{s_k^T y_k} - \frac{s_k^T y_k}{\|s_k\|^2}\right)v_k^2 + \left(\tau_k + \frac{s_k^T y_k}{\|s_k\|^2}\right)v_k - 1 = 0, \tag{23}$$

which is equivalent to a quadratic equation in terms of v_k. As a special case, for $\tau = c = 0$, we have $\tau_k = 0$ and the solution of (23) is reduced to v_k^{OL} in (13), proposed in [34]. For the general case, the solution of (23) is

$$v_k^{\pm} = \frac{-\left(\tau_k + \frac{s_k^T y_k}{\|s_k\|^2}\right) \pm \sqrt{\left(\tau_k + \frac{s_k^T y_k}{\|s_k\|^2}\right)^2 + 4\tau_k\left(\frac{\|y_k\|^2}{s_k^T y_k} - \frac{s_k^T y_k}{\|s_k\|^2}\right)}}{2\tau_k\left(\frac{\|y_k\|^2}{s_k^T y_k} - \frac{s_k^T y_k}{\|s_k\|^2}\right)}. \tag{24}$$

Now, similar to [10], we consider the following definition

$$\bar{M}_k = \max\left\{\varepsilon, \min\left\{\frac{1}{\varepsilon}, \frac{\|y_k\|^2}{s_k^T y_k} - \frac{s_k^T y_k}{\|s_k\|^2}\right\}\right\}, \tag{25}$$

where ε is a small positive constant. The truncated version of (24) is as follows

$$v_k^{\pm} = \frac{-B_k \pm \sqrt{B_k^2 + 4A_k}}{2\tau_k \bar{M}_k}, \tag{26}$$

where $A_k = \tau_k\left(\frac{\|y_k\|^2}{s_k^T y_k} - \frac{s_k^T y_k}{\|s_k\|^2}\right)$ and $B_k = \tau_k + \frac{s_k^T y_k}{\|s_k\|^2}$. The well definedness of (26) is shown in Lemma 1.

Lemma 1. *The roots $v_k^{\pm}$ in (26) of the quadratic equation (23) are real and well defined, provided that $\tau_k > 0$ and $s_k^T y_k > 0$.*

Proof. Based on the Cauchy–Schwarz inequality, it is clear that

$$(s_k^T y_k)^2 \leq \|s_k\|^2 \|y_k\|^2. \tag{27}$$

By dividing the both sides of (27) to $s_k^T y_k > 0$ and rewriting (27), we have

$$\frac{\|y_k\|^2}{s_k^T y_k} - \frac{s_k^T y_k}{\|s_k\|^2} > 0,$$

provided that s_k and y_k are independent vectors. Now, by multiplying by $\tau_k > 0$, we have

$$4\tau_k\left(\frac{\|y_k\|^2}{s_k^T y_k} - \frac{s_k^T y_k}{\|s_k\|^2}\right) > 0. \tag{28}$$

By (28) we have

$$B_k^2 + 4A_k = \left(\tau_k + \frac{s_k^T y_k}{\|s_k\|^2}\right)^2 + 4\tau_k\left(\frac{\|y_k\|^2}{s_k^T y_k} - \frac{s_k^T y_k}{\|s_k\|^2}\right) > 0,$$

which shows that $v_k^{\pm}$ in (26) are real. Moreover to indicate the well definednees, we shown that the denominator (26) has a positive lower bound. Let there exist a constant $\varepsilon_2 > 0$, $\|g_k\| \geq \varepsilon_2$, otherwise the convergent holds. So from (8), the inequality $|\tau_k| \geq C\|g_k\|^P \geq C\varepsilon_2^P$ is holds and from (25), $\bar{M}_k \geq \varepsilon$, which leads to $2\tau_k \bar{M}_k \geq 2C\varepsilon_2^P \varepsilon$. □

To set the new scale parameter, based on (26) and Lemma 1, we can define

$$v_k^{DT} = \max\{v_k^+, v_k^-\} > 0, \tag{29}$$

which guarantees the positiveness. Since $v_k^+ \geq v_k^-$, therefore from (29), we have $v_k^{DT} = v_k^+$. In the third approach of setting the parameter v_k in (14) or (15), the measure function is applied. The main advantage of this method is the use of both the trace and the determinant of the search direction matrix. Byrd and Nocedal [15] proposed the measure function

$$\varphi(B_{k+1}) = \mathrm{tr}(B_{k+1}) - \ln(\det(B_{k+1})), \tag{30}$$

where $\ln(\cdot)$ denotes the natural logarithm. Since B_{k+1} is a positive definite matrix, it follows that the measure function (30) is well defined. However, it is quite possible that in some iterations we have $\ln(\det(B_{k+1})) < 0$. This is more harmful to minimiz $\varphi(B_{k+1})$ in (14). Therefore, here another measure function given by Dennis and Wolkowicz [20] is applied, which is defined as follows

$$w(B_{k+1}) = \frac{\mathrm{tr}(B_{k+1})}{n\big(\det(B_{k+1})\big)^{\frac{1}{n}}}.$$

Now, by replacing the trace and the determinant of B_{k+1}, given in (19) and (22), respectively, we have a new function, which depends on v_k and should be minimized, similar to [5] for SMBFGS. Therefore, the new optimization problem $\min\limits_{v_k>0} w(B_{k+1})$ should be solved. After some manipulations, using the optimal necessary condition, this problem can be converted to the quadratic algebraic equation

$$\begin{aligned}
&\tau_k(n-2)\big(\frac{\|y_k\|^2}{s_k^T y_k} - \frac{s_k^T y_k}{\|s_k\|^2}\big)\big(\frac{\|y_k\|^2}{s_k^T y_k} + \tau_k\big)v_k^2 \\
&+(n-1)\Bigg(\big(\tau_k + \frac{s_k^T y_k}{\|s_k\|^2}\big)\big(\frac{\|y_k\|^2}{s_k^T y_k} + \tau_k\big) - 2\tau_k\big(\frac{\|y_k\|^2}{s_k^T y_k} - \frac{s_k^T y_k}{\|s_k\|^2}\big)\Bigg)v_k - \big(\tau_k + \frac{s_k^T y_k}{\|s_k\|^2}\big)(n-1) = 0,
\end{aligned} \tag{31}$$

by solving $\frac{dw}{dv_k} = 0$. In the special case when $\tau = c = 0$, we have $\tau_k = 0$ and the solution of (31) is reduced to v_k, proposed by Oren and Spedicato [35] in (13). Similar to (26), the truncated version of the solution of (31) is as follows

$$v_k^{\pm} = \frac{-N(B_kC_k - 2A_k) \pm \sqrt{(B_kC_kN)^2 + 4A_k^2N^2 - 4A_kB_kC_kN}}{2(N-1)\bar{M}_kC_k}, \tag{32}$$

where, $C_k = \frac{\|y_k\|^2}{s_k^T y_k} + \tau_k$, $N = n-1$. The well definedness of (32) is shown in Lemma 2.

Lemma 2. *The roots $v_k^{\pm}$ in (32) of the quadratic equation (31) are real and well defined, provided that $\tau_k > 0$ and $s_k^T y_k > 0$.*

Proof. Since $s_k^T y_k > 0$, then $A_k = \tau_k\big(\frac{\|y_k\|^2}{s_k^T y_k} - \frac{s_k^T y_k}{\|s_k\|^2}\big) \neq 0$ and $C_k = \frac{\|y_k\|^2}{s_k^T y_k} + \tau_k \neq 0$. Therefore, $v_k^{\pm}$ well defined. Also, similar to Lemma 1, it can be seen that $v_k^{\pm}$ are real. We define v_k^{MF} as follows:

$$v_k^{MF} = \max\{v_k^+, v_k^-\} = v_k^+ > 0. \tag{33}$$

By replacing the spectral parameter v_k in direction (16) with one of the proposed parameters v_k^{TR} in (21), v_k^{DT} in (29), or v_k^{MF} in (33), the new SMABFGS update, called NSMA, is obtained, which is summarized in Algorithm 1. □

Algorithm 1 NSMA algorithm.

Input Choose an initial point $x_0 \in \mathbb{R}^n$, the constants $0 < \delta < \sigma < 1$, $\tau, C, p \in \mathbb{R}$, and $0 < \varepsilon < 1$ sufficiently small. Set $g_0 = \nabla f(x_0)$, $d_0 = -g_0$.
Step 1. If $\|g_k\| \leq \varepsilon$, then stop.
Step 2. Compute the step size $\alpha_k > 0$, satisfying the Wolfe LS conditions (3)–(4).
Step 3. Compute $x_{k+1} = x_k + \alpha_k d_k$, $f_{k+1} = f(x_{k+1})$, and $g_{k+1} = \nabla f(x_{k+1})$. Then set $s_k = x_{k+1} - x_k$ and $y_k = g_{k+1} - g_k$, and compute the scaling factors v_k^* by one of (21), (29), (33).
Step 4. Compute the search direction by (16) with $v_k = v_k^*$.
Step 5. Set $k = k+1$, and go to Step 1.

Based on selecting the scaled parameter in Step 3 of Algorithm 1, we have three versions of the NSMA algorithm, shown by TR, DT, and MF methods.

3 Convergence analysis

In this section, the convergence analysis of Algorithm 1 is proved for the uniformly convex and general functions, separately. First of all, we need to assume some basic assumptions on the objective function as follows.

Assumption 1. *For arbitrary $x_0 \in \mathbb{R}^n$, suppose that $S = \{x \in \mathbb{R}^n | f(x) \leq f(x_0)\}$ is a bounded set, that is, there exists a constant $a > 0$ such that*

$$\|x\| \leq a, \qquad \textit{for all } \; x \in S. \tag{34}$$

In a neighborhood N of S, $\nabla f(x)$ is Lipschitz continuous, that is, there exists a constant $L > 0$ such that

$$\|\nabla f(x) - \nabla f(y)\| \leq L\|x - y\|, \qquad \textit{for all } \; x, y \in N. \tag{35}$$

Based on Assumption 1, there exists a positive constant μ such that

$$\|\nabla f(x)\| \leq \mu, \qquad \text{for all } \; x \in S. \tag{36}$$

From inequalities (34) and (36), it can be shown that, there exist a positive constant $\tilde{M}$ such that

$$|f(x)| \leq \tilde{M}, \quad \text{for all } \; x \in S. \tag{37}$$

Moreover from (35), it results in

$$\|y_k\| \leq L\|s_k\|. \tag{38}$$

If a smooth function f is uniformly convex on S, then there exists a constant $\zeta > 0$ such that

$$y_k^T s_k \geq \zeta \|s_k\|^2, \qquad \text{for all } \; k \geq 0. \tag{39}$$

The boundedness of the parameter v_k in (14) or (15) is important. This issue has been proved for the parameter v_k^{OS} in [2]. Now, we prove it in the following Lemma for the proposed parameter v_k^* in (21), (29), and (33).

Lemma 3. *If f is a uniformly convex function on a neighborhood N of S, then the proposed scaled parameters v_k^{TR}, v_k^{DT}, and v_k^{MF} are bounded.*

Proof. First for v_k^{TR}, since $s_k^T y_k > 0$, for $\tau_k > 0$, we have

$$\|y_k\|^2 + \tau_k s_k^T y_k > \|y_k\|^2. \tag{40}$$

Now, using the definition of v_k^{TR} in (21) and the inequality (40), we obtain

$$|v_k^{TR}| = \left|\frac{s_k^T y_k}{\|y_k\|^2 + \tau_k s_k^T y_k}\right|.$$

Using the Cauchy–Schwarz inequality, $s_k^T y_k \leq \|s_k\|\|y_k\|$, (39), and (40), we have

$$|v_k^{TR}| \leq \frac{\|s_k\|\|y_k\|}{\|y_k\|^2} = \frac{\|s_k\|}{\|y_k\|} \leq \frac{1}{\zeta}, \tag{41}$$

which shows that the parameter v_k^{TR} is bounded. To show the boundedness of v_k^{DT}, from the mean theorem and (8), (35), (36), and the Cauchy–Schwarz inequality, we have

$$|\theta_k| = |2(f_k - f_{k+1}) + s_k^T(g_k + g_{k+1})| = \left|\big(-2\nabla f(x_z) + \nabla f(x_k) + \nabla f(x_{k+1})\big)^T s_k\right|,$$

where, $x_z = zx_k + (1-z)x_{k+1}$, for some $z \in (0,1)$. Therefore

$$\begin{aligned} |\theta_k| &\leq (\|\nabla f(x_k) - \nabla f(x_z)\| + \|\nabla f(x_{k+1}) - \nabla f(x_z)\|)\|s_k\| \\ &\leq (L(1-z)\|s_k\| + Lz\|s_k\|)\|s_k\| = L\|s_k\|^2, \end{aligned} \tag{42}$$

where, L is a positive constant. Moreover,

$$|\tau_k| = \left|\frac{\tau_k\theta_k}{\|s_k\|^2} + C\|g_k\|^p\right| \leq \tau\frac{L\|s_k\|^2}{\|s_k\|^2} + C\|g_k\|^P \leq \tau L + C\mu^P = L_1, \tag{43}$$

and

$$\left|\frac{s_k^T y_k}{\|s_k\|^2}\right| \leq \frac{\|s_k\|\|y_k\|}{\|s_k\|^2} = \frac{\|y_k\|}{\|s_k\|} \leq L. \tag{44}$$

From (43) and (44), the inequality

$$|B_k| = \left|\tau_k + \frac{s_k^T y_k}{\|s_k\|^2}\right| \leq L_1 + L = M_1, \quad |C_k| = \left|\frac{\|y_k\|^2}{s_k^T y_k} + \tau_k\right| \leq \frac{L^2}{\zeta} + L_1 = M_2, \tag{45}$$

is satisfied. Moreover

$$|A_k| = |\tau_k|\left|\frac{\|y_k\|^2}{s_k^T y_k} - \frac{s_k^T y_k}{\|s_k\|^2}\right| \leq L_1\left(\frac{\|y_k\|^2}{\zeta\|s_k\|^2} + \frac{\|s_k\|\|y_k\|}{\|s_k\|^2}\right) \leq \left(\frac{L^2}{\zeta} + L\right)L_1 = M_3. \tag{46}$$

For the lower bound of the denominator of v_k^{DT} in (29), assuming that there exist a constant $\varepsilon_2 > 0$, $\|g_k\| \geq \varepsilon_2$, otherwise the convergent holds, we have $|\tau_k| \geq C\varepsilon_2^P$. Now, using (25)

$$2|\tau_k|\bar{M}_k \geq 2C\varepsilon_2^P \varepsilon. \tag{47}$$

From (45), (46), (47), and (29) we conclude that

$$|v_k^{DT}| = \left|\frac{-B_k + \sqrt{B_k^2 + 4A_k}}{2\tau_k \bar{M}_k}\right| \leq \frac{(L_1 + L) + \sqrt{(L_1 + L)^2 + 4(\frac{L^2}{\zeta} + L)L_1}}{2C\varepsilon_2^P \varepsilon}. \tag{48}$$

On the other hand, for the lower bound of v_k^{DT}, by rationalizing the numerator of v_k^{DT} in (29) and inequalities (45), (46) and (47), we have:

$$\begin{aligned}\left| -B_k + \sqrt{B_k^2 + 4A_k}\right| &= \left|\frac{4A_k}{B_k + \sqrt{B_k^2 + 4A_k}}\right| = \left|\frac{4\tau_k \bar{M}_k}{B_k + \sqrt{B_k^2 + 4A_k}}\right| \\ &\geq \frac{4C\varepsilon_2^P \varepsilon}{(L_1 + L) + \sqrt{(L_1 + L)^2 + 4(\frac{L^2}{\zeta} + L)L_1}} = m_1.\end{aligned}$$

Also, from $\bar{M}_k \leq \frac{1}{\varepsilon}$ and (43) it is clear that $|2\tau_k \bar{M}_k| \leq \frac{2L_1}{\varepsilon}$. Therefore, we have

$$|v_k^{DT}| = \left|\frac{-B_k + \sqrt{B_k^2 + 4A_k}}{2\tau_k \bar{M}_k}\right| \geq \frac{\varepsilon m_1}{2L_1},$$

which can be considered as a lower bound for v_k^{DT}. Finally, we prove the boundedness of v_k^{MF}. From the defination of v_k^{MF} in (33) we have

$$|v_k^{MF}| = \frac{-B_{1k} + \sqrt{(B_k C_k N)^2 + 4A_k^2 N^2 - 4C_{1k}}}{2(N-1)\bar{M}_k C_k}, \tag{49}$$

where, $B_{1k} = (n-1)B_k C_k - 2A_k$, $C_{1k} = (n-1)A_k B_k C_k$.
In following, we obtain the lower and upper bound of (49). For the upper bound, we show that the numerator of (49) has as upper bound and the denumerator of it has lower bound. Since

$$|B_{1k}| = (n-1)|B_k C_k - 2A_k| \leq (n-1)(|B_k C_k| + 2|A_k|) \leq (n-1)(M_1 M_2 + 2M_3) = M_4, \tag{50}$$

and

$$|C_{1k}| = (n-1)|A_k B_k C_k| \leq (n-1)M_1 M_2 M_3 = M_5, \tag{51}$$

therefore

$$\left| -B_{1k} + \sqrt{(B_k C_k N)^2 + 4A_k^2 N^2 - 4C_{1k}}\right| \leq M_4 + \sqrt{M_1 M_2 + 4M_3^2 + 4M_5}. \tag{52}$$

Moreover from (26) and (32), we have

$$|\tau_k| \geq C\varepsilon_2^P, \quad |C_k| \geq |\tau_k| \geq C\varepsilon_2^P, \quad |B_k| \geq |\tau_k| \geq C\varepsilon_2^P, \tag{53}$$

and

$$2(N-1)|\bar{M}_k C_k| \geq 2(N-1)|\bar{M}_k||\tau_k| \geq 2(N-1)C\varepsilon_2^P \varepsilon = M_6. \tag{54}$$

Therefore, from (52) and (54), we have

$$|v_k^{MF}| = \frac{\left| -B_{1k} + \sqrt{(B_k C_k N)^2 + 4A_k^2 N^2 - 4C_{1k}} \right|}{2(N-1)|\bar{M}_k C_k|} \leq \frac{M_4 + \sqrt{M_1 M_2 + 4M_3^2 + 4M_5}}{M_6}, \tag{55}$$

which is an upper bounded for parameter v_k^{MF}. Similar to the upper bound process for the lower bound of v_k^{MF} for lower bound of v_k^{MF}, by rationalizing the numerator of v_k^{MF} in (33) and inequalities (52), (53) and (54), we have

$$\begin{aligned} \left| -B_{1k} + \sqrt{(B_k C_k N)^2 + 4A_k^2 N^2 - 4C_{1k}} \right| &= \frac{\left| 4(N-1)\bar{M}_k B_k C_k \right|}{\left| B_{1k} + \sqrt{(B_k C_k N)^2 + 4A_k^2 N^2 - 4C_{1k}} \right|} \\ &\geq \frac{4(N-1)C^2 \varepsilon_2^{2P} \varepsilon}{M_4 + \sqrt{M_1 M_2 + 4M_3^2 + 4M_5}} = m_2. \end{aligned}$$

On the other hand, since

$$\bar{M}_k \leq \frac{1}{\varepsilon}, \quad |C_k| \leq \frac{L^2}{\zeta} + L_1,$$

it is clear that

$$|\tau_k \bar{M}_k C_k| < \frac{L_1}{\varepsilon}\left(\frac{L^2}{\zeta} + L_1\right) = L_2.$$

So, we have

$$|v_k^{MF}| = \left| \frac{-B_{1k} + \sqrt{(B_k C_k N)^2 + 4A_k^2 N^2 - 4C_{1k}}}{2(N-1)\bar{M}_k C_k} \right| \geq \frac{m_2}{2(N-1)L_2},$$

which is a lower bound of v_k^{MF}. □

By Lemma 3, the scaling parameters v_k^{TR}, v_k^{DT}, and v_k^{MF} are bounded, that is,

$$v_k^{TR}, v_k^{DT}, v_k^{MF} \in [m, M], \tag{56}$$

Lemma 4. *If f is uniformly convex on a neighborhood N of S, then directions (2) with v_k^{TR}, v_k^{DT}, and v_k^{MF} enforce the sufficient descent condition, that is $g_k^T d_k \leq -\eta \|g_k\|^2$, where $\eta > 0$ is constant.*

Proof. Similar to the proof of Lemma 3.6 of [8], it suffices to prove that $tr(B_{k+1})$ is bounded above. From (11), (35), (36), (43) and (44), we have

$$tr(B_{k+1}) \leq \frac{n-1}{v_k} + \frac{L^2}{\zeta} + \tau \frac{\|\theta_k\|}{\|s_k\|^2} + C\|g_k\|^p \leq \frac{n-1}{m} + \frac{L^2}{\zeta} + \tau L + C\mu^p. \qquad \square$$

Now, to prove the convergence of the NSMA, we need to use the following result.

Lemma 5. *[36] Suppose that Assumption 1 holds. Consider any LS method, where d_k satisfies the sufficient condition (2) and the Wolfe LS conditions, (3)–(4). If*

$$\sum_{k=0}^{\infty} \frac{1}{\|d_k\|^2} = \infty,$$

then the method converges globally in the sense that

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

Theorem 1. *If f is uniformly convex on a neighborhood N of S, then the NSMA algorithm with v_k^{TR}, v_k^{DT}, and v_k^{MF} parameters is converged.*

Proof. Lemma 4 implies that $d_k \neq 0$ for all $k \geq 0$. Hence considering Lemma 5, it suffices to prove that d_k in (16) is bounded above. In this context, from (10), (39), and the Cauchy–Schwarz inequality, we have

$$\gamma_k \geq \frac{s_k^T y_k}{\|s_k\|^2} \geq \zeta.$$

Given $\bar{K} = 1 + \frac{ML^2}{\zeta}$, the above inequalities together with (10), (12), (36), and (39) lead to

$$\begin{aligned}\|H_{k+1}\| \leq & v_k + 2v_k \frac{\|s_k\|\|y_k\|}{s_k^T y_k} + \left(1 + v_k \frac{\|y_k\|^2}{s_k^T y_k}\right) \frac{\|s_k\|^2}{s_k^T y_k} + \frac{\tau_k}{\gamma_k} \frac{\|z_k\|^2}{s_k^T y_k} \\ \leq & M + 2M\frac{L}{\zeta} + \frac{1}{\zeta}\bar{K} + \frac{t\xi + C\mu^p}{\zeta^2}\left(LM + \bar{K}\right)^2 = \Lambda.\end{aligned}$$

Therefore, it follows from (2) and (36) that

$$0 < \|d_{k+1}\| \leq \|H_{k+1}\|\|g_{k+1}\| \leq \Lambda\mu.$$

So, the sequence of search direction $\{d_k\}_{k\geq 0}$ is bounded above. Now, from Lemma 5, the global convergence is guaranteed. $\square$

For the general function, the global convergence of the proposed NSMA method can be achieved as in [2], in which the vector y_k in (10) is replaced by $\tilde{y}_k$ in (7); for more detail, see [2, 44].

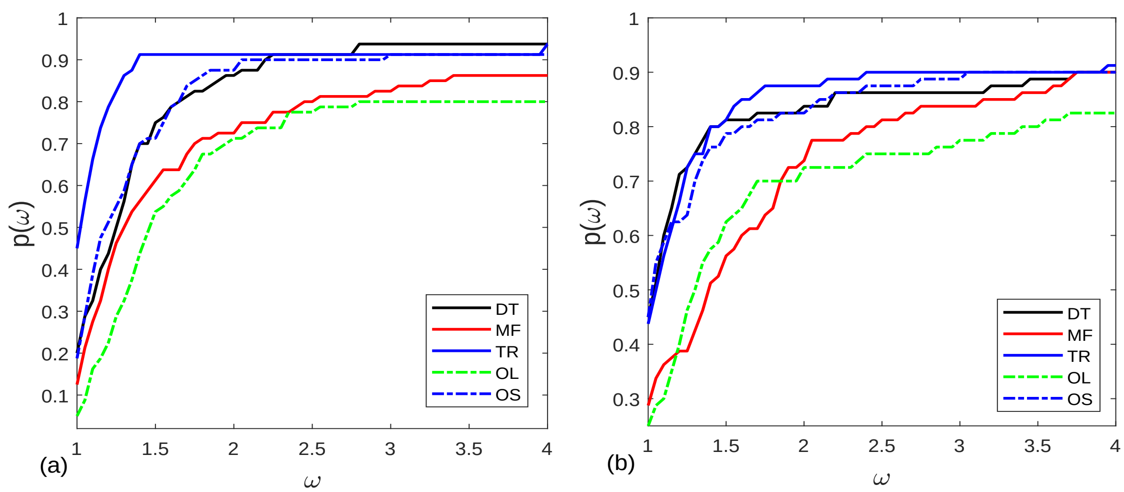


Figure 1: Dolan–More performance profiles based on CPUT (a) and TNF (b).

4 Numerical experiments

In this section, to investigate the efficiency of the proposed method (NSMA algorithm or Algorithm 1), two numerical experiments are implemented. The first is based on the collection of CUTEr test problems [23], and the second is based on image processing as an applicable case study. Moreover, to compare the proposed scale parameters, we consider v_k^{TR} in (21), v_k^{DT} in (29), v_k^{MF} in (33), v_k^{OS} in (13), and v_k^{OL} in (13) in the NSMA algorithm. For the LS procedure, the Wolfe LS conditions described in (3)–(4) with the parameters $\sigma = 0.99$ and $\delta = 10^{-4}$ as in [2] are considered. All algorithms are stopped if $\|g\|_\infty < 10^{-6}$ the number of iterations exceed 10000. The parameters of the NSMA algorithm are chosen as $p = 1$, $\tau = 1$, and $C = 10^{-3}$, Moreover, similar to [10], the parameter in (25) is set to 10^{-8}. The codes were written in MATLAB 9.4.0.8 (R2018).

4.1 CUTEr collection

In this subsection, a set of 80 UO test problems from the CUTEr collection [23] is selected. In Table 1, the first column is the name of the problems; the second column is the dimension of them, which vary from 50 to 10000; also, for TR, DT, and MF methods, the third, fourth, and fifth columns, respectively, represent the CPU time (CPUT), number of iteration (NI), and the total number of evaluations of function (NF) and gradient (NG), which is defined by TNF=NF+3NG.

To approximately assess the performance of different algorithms, we use the performance profile introduced by Dolan and More [21] with respect to CPUT and the weighted sum for NF and NG as TNF.

Figure 1 shows the performances of the NSMA algorithm, which contains TR, DT, MF, OL, and OS methods with respect to CPUT and TNF criteria.

Comparing the methods shown in Figure 1, we can see that the TR and DT methods performed better than the MF, OL, and OS methods. Moreover, MF and OS methods are competitive, and that all methods are better than the method OL.

Table 1: The test problems

Function	n	$\frac{TR}{TIME/NI/TNF}$	$\frac{DT}{TIME/NI/TNF}$	$\frac{MF}{TIME/NI/TNF}$
ARGLINA	200	1.45E-01/ 2/ 12	1.92E-01/ 2/ 12	2.00E-01/ 2 /12
BDEXP	5000	8.83E-02/ 2/ 12	9.86E-02/ 2 /12	2.86E-01/ 55/ 233
BDQRTIC	5000	5.56E-01/ 162/ 1086	9.94E-01/ 370/ 985	1.24E-01/ 460/ 1012
BIGGSB1	5000	2.05E+00/ 2091/ 9442	2.98E+00/ 2596/ 1245	5.19E+00/ 4235 /12456
BOX	10000	2.54E-01/ 8/ 58	7.00E-01/ 89 /705	3.04E-01/ 15 /106
BROWNAL	200	7.14E-02 /18/ 94	8.03E-02/ 30 /169	5.12E-02/ 4/ 24
BROYDN7D	5000	5.19E+01 /8042/ 32780	5.95E-01/ 9326 /22567	6.90E-01/ 10000/ 31234
BRYBND	5000	2.57E-01/ 45 /239	7.12E-01 /244 /1097	5.74E-01/ 133 /852
CHAINWOO	4000	1.67E-01/ 10000/ 42028	1.64E-02/ 10000/ 39871	8.03E-01 /393/ 1671
COSINE	10000	1.43E-01 /4 /33	1.26E-01/ 4/ 33	2.14E-01 /14 /79
DIXMAANA	3000	5.99E-02 /6/ 31	7.60E-02/ 6 /31	2.93E-01 /107/ 76
DIXMAANB	3000	6.09E-02 /6 /31	5.86E-02/ 6/ 31	3.00E-01/ 121 /55
DIXMAANC	3000	5.28E-02/ 6/ 32	6.97E-02/ 6 /32	2.58E-01 /116/ 65
DIXMAAND	3000	5.79E-02/ 7 /37	6.59E-02 /7/ 37	6.29E-02/ 7/ 37
DIXMAANE	3000	4.04E-01/ 339 /1459	6.62E-01/ 593 /1345	5.64E-01 /369 /1554
DIXMAANF	3000	3.43E-01/ 269/ 1185	4.15E-01 /326 /975	5.83E-01 /414 /1748
DIXMAANG	3000	2.32E-01 /173 /752	2.94E-01/ 247 /987	3.92E-01/ 305/ 1314
DIXMAANH	3000	2.39E-01 /166/ 726	4.45E-01 /349 /1441	5.69E-01/ 461/ 1903
DIXMAANI	3000	8.04E-01/ 685 /3038	9.00E-01 /812/ 1280	2.02E-01/ 1165/ 2567
DIXMAANJ	3000	2.36E-01/ 173/ 761	2.37E-01 /177/744	2.38E-01 /177 /734
DIXMAANK	3000	1.64E-01/ 106/ 458	1.69E-01 /119 /490	2.07E-01 /147 /609
DIXMAANL	3000	1.27E-01/ 76 /338	2.34E-01/ 174 /734	1.91E-01 /127/525
DQDRTIC	5000	1.42E-01/ 37 /189	1.81E-01 /60 /157	1.97E-01/ 63 /250
DQRTIC	5000	3.41E-02/ 0 /4	5.06E-02 /0 /4	4.25E-02/ 0 /4
ENGVAL1	5000	1.09E-01/ 10 /50	1.30E-01/ 10 /50	1.45E-01 /24/ 138
EXTROSNB	1000	7.84E-01/ 1370 /7570	1.98E-01/ 3340/ 20990	7.28E+01 /10000 /40440
FLETCBV2	5000	8.96E-02/ 0 /4	9.32E-02/ 0 /4	1.10E-01 /0 /4
GENROSE	500	7.10E-01/ 1496/ 6636	8.77E-01 /1617 /6655	8.58E-01 /1713/ 7081
LIARWHD	5000	2.77E-01 /101/ 687	5.16E-01/ 198/ 1617	2.42E-01 /97 /323
NONDIA	5000	1.47E-01 /45 /328	5.64E-01 /133 /885	1.96E-01 /72 /447
PENALTY2	200	8.69E-03/ 0 /4	1.21E-02/ 0/ 4	2.09E-02/ 0/ 4
QUARTC	5000	3.40E-02/ 0 /4	4.47E-02/ 0 /4	4.11E-02 /0/ 4
SCHMVETT	5000	1.65E-01/ 9/ 47	1.46E-01/ 9 /47	1.64E-01/ 12/ 64
SPARSQUR	10000	4.52E-01/ 83 /348	4.94E-01 /88/ 250	4.70E-01 /81/ 361
SPMSRTLS	4999	6.71E-01 /278 /1191	1.48E+00 /663 /765	7.93E-01 /339 /1415
SROSENBR	5000	9.00E-02 /26 /183	1.25E-01 /53 /336	9.81E-02/ 26 /153
TOINTGSS	5000	1.03E-01/ 9 /46	1.01E-01/ 9 /46	1.09E-01/ 9/ 46
TRIDIA	5000	2.85E+00/ 2916/ 13110	5.54E+00/ 5244/ 1508	4.25E+00 /3985 /16660
VARDIM	200	7.56E-03 /1 /8	8.78E-03 /1 /8	5.47E-02/ 1/ 8
VAREIGVL	50	2.31E-02/ 24/ 113	3.22E-02/ 23 /110	4.69E-02/ 33 /153
WOODS	4000	6.93E-02/ 20 /94	7.85E-02/ 24/ 109	3.41E-01/ 226 /256
CHNROSNB	50	1.78E-01/ 399/ 1759	2.04E-01 /472 /1995	2.06E-01 /538 /1986
CRAGGLVY	5000	4.36E-01/ 74 /316	3.86E-01/ 65 /282	5.18E-01 /85/ 381
CURLY10	10000	3.00E-01/ 47 /235	3.02E-01/ 46 /232	3.60E-01 /72 /350
CURLY20	10000	6.40E-01/ 105/ 499	5.95E-01/ 84 /384	7.00E-01 /126 /563
CURLY30	10000	1.00E+00/ 164 /774	8.39E-01 /113 /504	1.01E+00/ 156/ 700
DECONVU	63	1.64E-01/ 407/ 2008	2.59E-01 /652 /1567	5.56E-01/ 1038/ 5287
EDENSCH	2000	6.38E-02/ 22/ 112	6.60E-02/ 22 /112	9.19E-02/ 45/ 195
EIGENBLS	2550	1.15E+01/ 7808/ 34772	1.40E-01/ 10000/ 33705	1.39E+01 /10000 /2456
ERRINROS	50	2.99E-01 /541/ 2625	1.50E+00 /3901/ 17692	4.03E+00 /10000/ 41205
FLETCHCR	1000	1.01E-01/ 98/ 508	1.33E-01 /129 /543	1.31E-01 /137 /657
FMINSRF2	5625	7.27E-01/ 369/ 1578	9.55E-01 /378 /1608	1.24E+00/ 568 /2395
FMINSURF	5625	1.17E+00 /550 /2442	1.40E+00 /626/ 2631	2.35E+00/ 1044 /4284
FREUROTH	5000	1.35E-01/ 15/ 99	1.45E-01 /15 /106	1.53E-01/ 28/ 162
GENHUMPS	5000	6.57E-02/ 0 /4	7.60E-02/ 0/ 4	5.98E-02/ 0/ 4
MANCINO	100	2.89E-01 /15 /93	2.90E-01 /15 /93	7.60E+00 /288 /3636
MOREBV	5000	1.02E-01/ 19/ 82	1.32E-01/ 32 /137	1.35E-01/ 37 /160
MSQRTALS	1024	1.05E+01/ 3097/ 13981	1.72E+01 /5397 /15432	1.18E+01 /3632/ 14936
MSQRTBLS	1024	7.59E+00/ 2255 /10108	1.24E+01/ 3763 /12345	7.89E+00/ 2375 /9863
NCB20	5010	4.97E-01 /63 /271	5.40E-01/ 62/ 282	1.43E+00/ 236 /961
NCB20B	5000	4.14E-01/ 43 /201	3.46E-016 40 /175	3.60E-01/ 40 /175
NONCVXU2	5000	4.41E-02 /0 /4	6.23E-02 /0 /4	4.28E-02 /0 /4
NONDQUAR	5000	1.90E+00 /1678/ 7708	3.10E+00 /2309 /4500	1.54E+00 /1401 /5783
POWELLSG	5000	1.96E-01 /147/ 717	2.16E-01 /139/ 624	1.92E-01/ 145/ 686
POWER	10000	1.10E+00 /674 /2965	1.51E+00 /878 /2650	1.89E+00 /1105/ 4483
SINQUAD	5000	2.71E-01 /9 /82	2.90E-01 /14/ 106	3.71E-01/ 28 /241
SPARSINE	5000	3.96E+01 /9870/ 45385	3.87E+01 /10000 /3500	3.88E+01 /10000 /41612
TESTQUAD	5000	1.02E+01/ 9938/ 45454	2.11E+01 /9765/ 3509	2.98E+01/ 9711 /42244
TOINTGOR	50	4.56E-02 /89 /373	4.93E-02/ 95 /398	4.93E-02 /84 /354
TOINTQOR	50	2.90E-02/ 23 /103	2.94E-02 /27 /119	2.65E-02 /24/ 106
TOINTPSP	50	5.53E-02/ 103/ 471	5.23E-02/ 115 /496	6.00E-02/ 118/ 522
TQUARTIC	5000	2.52E-01/61/ 455	2.18E-01/53 /346	1.95E-01/ 45 /273
CHENHARK	5000	1.88E-01/ 98/ 520	9.68E+00 /8566/ 1789	4.22E-01 /284 /1172
CLPLATEB	5041	2.09E+01 /10000 /45272	2.13E+01 /10000 /25600	2.10E+01 /10000 /2678
DRCAV1LQ	4489	1.20E-01 /0/ 4	8.66E-02 /0/ 4	1.46E-01 /0 /4
DRCAV2LQ	4489	1.14E-01 /0/ 4	1.23E-01/ 0 /4	8.34E-02/ 0/ 4
DRCAV3LQ	4489	1.08E-01 /0 /4	1.00E-01/ 0 /4	1.01E-01 /0 /4
FLETCBV3	5000	1.94E-01/ 3 /164	1.80E-01/ 3 /162	1.51E-01 /3/ 160
FLETCHBV	5000	1.92E-01 /3 /164	1.95E-01 /3 /164	1.53E-01/ 3/ 156
GENHUMPS	5000	6.45E-02 /0/ 4	1.02E-02 /0/4	5.84E-02 /0/ 4

4.2 Image processing

Digital image processing is widely used in the subject literature, for example, in medicine, photography, security, and so on; see [27, 42]. Noise problem-solving methods have been used in [12]. In image processing, one of the main tasks is to remove noise. A filter is used to reduce the amount of unwanted noise in a specific image, and a filter usually operates on a neighborhood of pixels in an image. There are various types of filters, such as average filter, median filter, and adaptive median filter. Also, types of noises can be called Gaussian noise, salt and pepper noise, and Poison noise. Yu et al. [41] introduced the following smooth function for image restoration:

$$f(x) = \sum_{(i,j)\in\mathcal{N}} \left\{ \sum_{(m,n)\in v_{i,j}\setminus\mathcal{N}} \varphi_\alpha(x_{ij} - \xi_{mn}) + \frac{1}{2} \sum_{(m,n)\in v_{i,j}\cap\mathcal{N}} \varphi_\alpha(x_{ij} - x_{mn}) \right\}, \tag{57}$$

where $\mathcal{N} = \{(i,j) \in A \setminus \bar{\xi}_{i,j} \neq \xi_{i,j}, \xi_{i,j} = s_{\max} \quad or \quad s_{\min}\}$, $v_{i,j} = \{(i,j-1),(i,j+1),(i-1,j),(i+1,j)\}$, and $A = \{1,2,\ldots,M\} \times \{1,2,\ldots,N\}$ is a neighborhood of (i,j). Moreover, ξ is the observed noisy image of X corrupted random by salt and pepper noise, $\bar{\xi}$ is obtained by applying median filter to the noisy image ξ, and $S_{\min}$ and $S_{\max}$, respectively, denote the minimum and maximum of a noisy pixel. Also, φ_α is an edge-preserving functional, which is chosen as $\varphi_\alpha(t) = \sqrt{t^2+\alpha}$. We set $\alpha = 0.75$ in our tests.

Image quality is measured by the three parameters: CPUT (in second), relative error (in short RElErr) (in percent), and peak signal-to-noise ratio (in short PSNR) (in dB) defined in [12]. The calculation formulas are as follows:

$$PSNR = 10\log_{10} \frac{255^2}{\frac{1}{M\times N}\|X^* - X\|}, \quad RELErr = 100 \times \frac{\|X^* - X\|}{\|X\|}.$$

An algorithm with large PSNR and small CPUT and RELErr is called efficient and chosen as the best algorithm. Three images named Bird, Lena, and Goldhill, used in [12], are chosen. Then we removed the noise by using salt and pepper noise with 35% degree in 256×256 dimensions. Applying TR, DT, and MF methods, we solve (57).

Table 2 shows the summary of these numerical results. Especially, this table demonstrates the superiority of TR, DT, and MF methods over OL and OS, and the best result of each line was shown in bold.

Table 2: Image restoration outputs: CPUT, RELErr, and PSNR.

		Method				
Critria	Figures	OL	OS	TR	DT	MF
CPUT	Bird	10.1242	10.1194	9.8818	9.8348	**9.7222**
	Lena	16.5207	16.2428	16.2234	16.2228	**16.2225**
	Goldhill	16.1247	16.3261	16.3424	**16.0481**	16.3905
RELErr	Bird	0.2695	0.2524	0.2521	0.2522	**0.2520**
	Lena	2.0798	2.0554	2.0536	2.0550	**2.0533**
	Goldhill	2.0739	2.0383	2.0377	**2.0272**	2.0274
PSNR	Bird	38.1461	38.3696	**38.3741**	**38.3741**	38.3740
	Lena	30.4665	30.6674	30.7751	30.7701	**30.7752**
	Goldhill	29.1855	29.5653	**29.5736**	**29.5735**	29.4734

Table 3: Images corrupted by 35% salt and pepper noise in the first line; the restored images via OL in the second line; the restored images via OS in the third line; the restored images via TR, DT and MF in the fourth, fifth, and sixth lines.

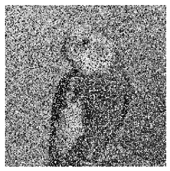
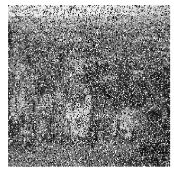
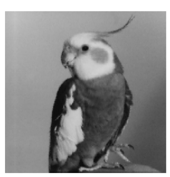
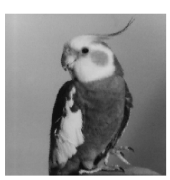
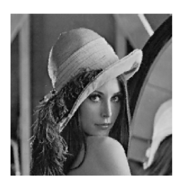
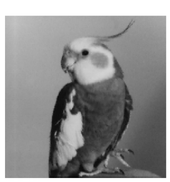
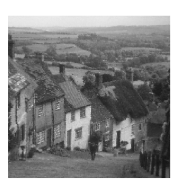
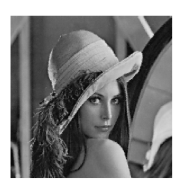
noise			
OL			
OS			
TR			
DT			
MF			

Table 3 shows the graphical results of Bird, Lena, and Goldhill images, for the OS, OL, TR, DT, and MF methods. Regarding the time criterion, we see that DT and MF methods have better performance than OL, OS, and TR methods. Compared to RELErr, the DT and MF methods perform better and compared to PSNR, the TR, DT, and MF methods perform better than OL and OS methods.

5 Conclusion

Due to the importance of effectively setting the spectral parameter in the SMBFGS algorithms and the key role of this parameter in the quality of the algorithms, we introduced new parameters in the augmented version of these methods. Assuming the idea of clustering, the eigenvalues about one point were applied to introduce the spectral parameter with three approaches based on the trace, the determinant, and the measure functions of the direction matrix. The condition of sufficient descent for the convex function and global convergence for the general function were proved. Our algorithm was efficient not only in the CUTEr collection but also it was able to remove noise.

References

[1] Z. Aminifard, S. Babaie-Kafaki, *Dai-Liao extensions of a descent hybrid nonlinear conjugate gradient method with application in signal processing,* Numer. Algorithms **89** (2021) 1369–1387.

[2] Z. Aminifard, S. Babaie-Kafaki, S. Ghafoori, *An augmented memoryless BFGS method based on a modified secant equation with application to compressed sensing*, Appl. Numer. Math. **167** (2021) 187–201.

[3] N. Andrei, *Eigenvalues versus singular values study in conjugate gradient algorithms for large-scale unconstrained optimization,* Optim. Methods Softw. **32** (2017) 534–551.

[4] N. Andrei, *Diagonal Approximation of the Hessian by Finite Differences for Unconstrained Optimization,* J. Optim. Theory Appl. **185** (2020) 859–879.

[5] N. Andrei, *New conjugate gradient algorithms based on self-scaling memoryless Broyden-Fletcher-Goldfarb-Shanno method,* Calcolo 57 (2020) 1–27.

[6] N. Andrei, *A double parameter self-scaling memoryless BFGS method for unconstrained optimization,* Comput. Appl. Math. **39** (2020) 159.

[7] M.R. Arazm, S. Babaie-Kafaki, R. Ghanbari, *An extended Dai-Liao conjugate gradient method with global convergence for nonconvex functions,* Matematicki **52** (2017) 361–375.

[8] S. Babaie-Kafaki, *A modified scaled memoryless BFGS preconditioned conjugate gradient method for unconstrained optimization,* 4OR 11 (2013) 361–374.

[9] S. Babaie Kafaki, *A modified scaling parameter for the memoryless BFGS updating formula,* Numer. Algorithms **72** (2016) 425–433.

[10] S. Babaie-Kafaki, Z. Aminifard, *Two parameter scaled memoryless BFGS methods with a nonmonotone choice for the initial step length,* Numer. Algorithms **82** (2019) 1345–1357.

[11] S. Babaie-Kafaki, Z. Aminifard, S. Ghafoori, *A hybrid quasi-Newton method with application in sparse recovery,* Comput. Appl. Math. **41** (2022) 249.

[12] S. Babaie-Kafaki, N. Mirhoseini, Z. Aminifard, *A descent extension of a modified Polak–Ribire–Polyak method with application in image restoration problem*, Optim. Lett. 17 (2023) 351–367.

[13] C.G. Broyden, *A class of methods for solving nonlinear simultaneous equations,* Math. Comput. **19** (1965) 577–593.

[14] C.G. Broyden, *The convergence of a class of double-rank minimization algorithms: 2. the new algorithm,* IMA J. Appl. Math. **6** (1970) 222–231.

[15] R.H. Byrd, J. Nocedal, *A tool for the analysis of quasi-Newton methods with application to unconstrained minimization,* SIAM J. Numer. Anal. **26** (1989) 727–739.

[16] R.H. Byrd, J. Nocedal, Y.-X. Yuan, *Global convergence of a class of Quasi-Newton methods on convex problems,* SIAM J. Numer. Anal. **24** (1987) 1171–1190.

[17] Y. H. Dai, C.-X. Kou, *A nonlinear conjugate gradient algorithm with an optimal property and an improved Wolfe line search,* SIAM J. Optim. **23** (2013) 296–320.

[18] J.E. Dennis, J.J. More, *A characterization of superlinear convergence and its application to quasi-Newton methods,* Math. Comput. **28** (1974) 549–560.

[19] J.E. Dennis, J.J. More, *Quasi-Newton methods, motivation and theory,* SIAM Review **19** (1977) 46–89.

[20] J.E. Dennis, H. Wolkowicz, *Sizing and least-change secant methods,* SIAM J. Numer. Anal. **30** (1993).

[21] E.D. Dolan , J.J. More, *Benchmarking optimization software with performance profiles,* Math. Program. **91** (2002) 201–213.

[22] L. Exl, J. Fischbacher, H. Oezelt, M. Gusenbauer, T. Schrefl, *Preconditioned nonlinear conjugate gradient method for micromagnetic energy minimization,* Comput. Phys Commun. **235** (2019) 179–186.

[23] N.I. Gould, D. Orban, P.L. Toint, *Cuter and sifdec: A constrained and unconstrained testing environment, revisited,* ACM Trans. Math. Softw. **29** (2003) 373–394.

[24] A. R. Heravi, G. Abed Hodtani, *A New Correntropy-Based Conjugate Gradient Backpropagation Algorithm for Improving Training in Neural Networks,* IEEE Trans. Neural Netw. Learn. Syst. **29** (2018) 6252–6263.

[25] D. H. Li, M. Fukushima, *A modified BFGS method and its global convergence in nonconvex minimization*, J. Comput. Appl. Math. **129** (2001) 15–35.

[26] M. Li, H. Liu, Z. Liu, *A new family of conjugate gradient methods for unconstrained optimization,* J. Appl. Math. Comput. 58 (2018) 219–234.

[27] W. Li, Y. Liu, J. Yang, W. Wu, *A new conjugate gradient method with smoothing $L_{1/2}$ regularization based on a modified secant equation for training neural networks*, Neural Process. Lett. **48** (2018) 955–978.

[28] A. Liao, *Modifying the BFGS method,* Oper. Res. Lett. **20** (1997) 171–177.

[29] J. Lin, C. Jiang, *An improved conjugate gradient parametric detection based on space-time scan,* Signal Process. **169** (2020) 107412.

[30] I.E. Livieris, V. Tampakas, P. Pintelas, *A descent hybrid conjugate gradient method based on the memoryless BFGS update,* Numer. Algorithms **79** (2018) 1169–1185.

[31] S. Nezhadhossein, *New nonlinear conjugate gradient methods based on optimal Dai-Liao parameters,* J. Math. Model. **8** (2020) 21–39.

[32] S. Nezhadhossein, *A modified descent spectral conjugate gradient method for unconstrained optimization,* Iran. J. Sci. Technol. Trans. A Sci. **45** (2021) 209–220.

[33] J. Nocedal, S. J. Wright, *Numerical Optimization,* Springer Series in Operations Research. Springer-Verlag, New York, 1999.

[34] S.S. Oren, D.G. Luenberger, *Self-scaling variable metric (ssvm) algorithms: Part i: Criteria and sufficient conditions for scaling a class of algorithms,* Manag. Sci. **20** (19774) 845–862.

[35] S.S. Oren, E. Spedicato, *Optimal conditioning of self-scaling variable metric algorithms,* Math. Program. **10** (1976) 70–90.

[36] K. Sugiki, Y. Narushima, H. Yabe, *Globally convergent three-term conjugate gradient methods that use secant conditions and generate descent search directions for unconstrained optimization,* J. Optim. Theory Appl. **153** (2012) 733–757.

[37] W. Sun, Y.-X. Yuan, *Optimization Theory and Methods: Nonlinear Programming,* Springer, New York, 2006.

[38] N. Ullah, J. Sabiu, A. Shah, *A derivative-free scaling memoryless Broyden-Fletcher-Goldfarb-Shanno method for solving a system of monotone nonlinear equations,* Numer. Linear Algebra Appl. **28** (2021) e2374.

[39] Z. Wei, G. Li, L. Qi, *New quasi-Newton methods for unconstrained optimization problems,* Appl. Math. Comput. **175** (2006) 1156–1188.

[40] R. Winther, *Some superlinear convergence results for the conjugate gradient method,* SIAM J. Numer. Anal. **17** (1980) 14–17.

[41] G. Yu, J. Huang, Y. Zhou, *A descent spectral conjugate gradient method for impulse noise removal,* Appl. Math. Lett. **23** (2010) 555–560.

[42] G. Yuan, J. Lu, Z. Wang *The PRP conjugate gradient algorithm with a modified WWP line search and its application in the image restoration problems,* Appl. Numer. Math. **152** (2020) 1–11.

[43] J.Z. Zhang, N.Y. Deng, L.H. Chen, *New quasi-Newton equation and related methods for unconstrained optimization*, J. Optim. Theory Appl. **102** (1999) 147–167.

[44] W. Zhou, L. Zhang, *A nonlinear conjugate gradient method based on the MBFGS secant condition,* Optim. Methods Softw. **21** (2006) 707–714.