

# A scaled BFGS method with two parameters for unconstrained optimization

Neculai Andrei<sup>1</sup>

May 5, 2017

*This paper is dedicated to Prof. Roman A. Polyak, a great contributor in both the theory and the applications of Nonlinear Programming. It is my great pleasure to write this paper in his honor on the occasion of his 80<sup>th</sup> birthday anniversary.*

**Abstract.** A scaled BFGS method with two parameters for unconstrained optimization is presented. In this method the first two terms of the known BFGS update formula are scaled with a positive parameter and the third one is scaled with another positive parameter. The parameter scaling the first two terms of the BFGS update is determined by clustering the eigenvalues of the scaled BFGS matrix. On the other hand, the parameter scaling the third term is determined as a preconditioner to the Hessian of the minimizing function combined with the minimization of the conjugacy condition from conjugate gradient methods. This parameter is determined to reduce the large eigenvalues, thus obtaining a better distribution of them. Under the inexact Wolfe line search, the global convergence of this scaled BFGS method with two parameters is proved in very general conditions, without assuming the convexity of the minimizing function. Using 80 unconstrained optimization test functions with a medium number of variables, the preliminary numerical experiments show that this scaled BFGS method is more efficient than the standard BFGS update or than some other scaled BFGS methods based on interpolation conditions: Biggs [M.C. Biggs, *Minimization algorithms making use of non-quadratic properties of the objective function*. Journal of the Institute of Mathematics and Its Applications 8, 315-327 (1971)] and Yuan [Y. Yuan, *A modified BFGS algorithm for unconstrained optimization*. IMA Journal Numerical Analysis 11, 325-332 (1991)], spectral scaling: Cheng and Li [W.Y. Cheng and D.H. Li, *Spectral scaling BFGS method*. Journal of Optimization Theory and Applications 146, 305-319 (2010)], scaling only the first two terms of the BFGS update: Nocedal and Yuan [J. Nocedal and Y. Yuan, *Analysis of self-scaling quasi-Newton method*. Mathematical Programming 61, 19-37 (1993)] or scaling the last terms of the BFGS update with two parameters: Liao [A. Liao, *Modifying BFGS method*. Operations Research Letters 20, 171-177 (1997)]. However, in this scaled BFGS method with two parameters, scaling the first two terms with a positive parameter for clustering the eigenvalues has only a marginal effect versus the scaling of the third term with diagonal preconditioning and conjugacy condition by Andrei [N. Andrei, *A double parameter scaled BFGS method for unconstrained optimization*. ICI Technical Report, Bucharest, May 2, 2017.].

**Keywords:** Unconstrained optimization . BFGS method . Scaled BFGS method . Global convergence . Numerical comparisons.

Mathematics Subject Classification (2010) 65K. 90C30

---

<sup>1</sup> Academy of Romanian Scientists, Center for Advanced Modeling and Optimization, 54 Splaiul Independenței, Sector 5, Bucharest, Romania.  
E-mail: nandrei@ici.ro

## 1. Introduction

For solving the unconstrained optimization problem

$$\min f(x), \quad (1.1)$$

where  $x \in R^n$ , one of the best methods is the BFGS introduced by Broyden [12], Fletcher [22], Goldfarb [25] and Shanno [44]. An excellent presentation of the theoretical aspects concerning the properties and the convergence of this method were given by Dennis and Moré [18, 19]. At the same time, a deep analysis of the BFGS method and its variants was presented by Nocedal [35]. The BFGS method is fast and robust and currently it is used in innumerable optimization software for solving unconstrained or constrained optimization problems.

The BFGS method consists of starting with any approximation to the Hessian matrix of the minimizing function  $f$ , and at each iteration, update this matrix by incorporating the curvature of the function measured along the step direction. Suppose that at the  $k$ -th iteration, a symmetric and positive definite matrix approximation  $B_k$  of the Hessian  $\nabla^2 f(x_k)$  is given and a search direction is computed as:

$$d_k = -B_k^{-1} g_k, \quad (1.2)$$

where  $g_k$  is the gradient of function  $f$  in point  $x_k$ ,  $g_k = \nabla f(x_k)$ .

The next iteration is computed as:

$$x_{k+1} = x_k + \alpha_k d_k, \quad (1.3)$$

where the stepsize  $\alpha_k$  satisfies the Wolfe line search conditions [52, 53]:

$$f(x_k + \alpha_k d_k) \leq f(x_k) + \sigma \alpha_k g(x_k)^T d_k, \quad (1.4)$$

$$g(x_k + \alpha_k d_k)^T d_k \geq \rho g(x_k)^T d_k, \quad (1.5)$$

the positive constants  $\sigma$  and  $\rho$  satisfying  $0 < \sigma < \rho < 1$ . The Hessian approximation is updated by:

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

$k = 0, 1, \dots$ , where  $s_k = x_{k+1} - x_k$ ,  $y_k = g_{k+1} - g_k$ , the initial approximation  $B_0$  being symmetric and positive definite.

An important property of the BFGS updating formula (1.6), which we call it standard BFGS, is that  $B_{k+1}$  inherits the positive definiteness of  $B_k$  if  $y_k^T s_k > 0$ . The condition  $y_k^T s_k > 0$  holds if the stepsize  $\alpha_k$  in (1.3) is determined by the Wolfe line search conditions (1.4) and (1.5). We note that the condition  $y_k^T s_k > 0$  is also guaranteed to hold if the stepsize  $\alpha_k$  is determined by the exact line search:  $\min\{f(x_k + \alpha d_k), \alpha > 0\}$ . Since  $B_k$  is positive definite, the search direction  $d_k$  generated by (1.2) is a descent direction of  $f$  at  $x_k$ , no matter whether the Hessian is positive definite or not.

In practical implementations the search direction is computed as

$$d_k = -H_k g_k, \quad (1.7)$$

where  $H_k$  is the BFGS approximation to the inverse Hessian  $\nabla^2 f(x_k)^{-1}$  of  $f$  at  $x_k$ , i.e.  $H_k = B_k^{-1}$ . With a little algebra, using the rank-one Sherman-Morrison-Woodbury formula twice, from (1.6) we get:

$$H_{k+1} = H_k - \frac{H_k y_k s_k^T + s_k y_k^T H_k}{y_k^T s_k} + \left(1 + \frac{y_k^T H_k y_k}{y_k^T s_k}\right) \frac{s_k s_k^T}{y_k^T s_k}. \quad (1.8)$$

Also, for the stepsize computation, in practical implementations the inexact Wolfe line search conditions (1.4) and (1.5) are used.

The main results concerning the convergence property of the BFGS method are as follows. For twice continuously differentiable convex functions with compact level sets, Powell [41] proved the global convergence of the BFGS algorithm. Under the exact line search or under some special inexact line searches, for convex minimization problems the BFGS method is globally convergent [14, 15, 20, 26, 40]. On the other hand, for nonconvex problems under the exact line search, Mascarenhas [34] proved that the BFGS method (and some other methods in the Broyden class) may fail. For non-convex functions with line searches that satisfy the Wolfe conditions, Yu-Hong Dai [17] showed that the BFGS method may fail. However, the BFGS method has very interesting properties and remains one of the most respectable quasi-Newton method for unconstrained optimization [23, 35].

The most important properties of the BFGS method are its *self-correcting quality* and *better corrections of the small eigenvalues than the large ones* (see Nocedal [35]). Concerning the self-correcting quality, it was proved that if the current inverse approximation to the Hessian  $H_k$  of the minimizing function incorrectly estimates the curvature of this function, i.e. if this estimate slows down the iteration, then the BFGS Hessian approximation will tend to correct itself within a few steps. Another important property of BFGS, explained by Nocedal [35], is that it better corrects small eigenvalues than large ones. Powell [43] proved that BFGS with inexact Wolfe line search is globally superlinear convergent for convex problems. On the other hand, Byrd and Nocedal [15] extended Powell's analysis and obtained global convergence of BFGS with backtracking line search. Furthermore, under the Wolfe inexact line search, Byrd, Nocedal and Yuan [14] established the global and the superlinear convergence of the Broyden's quasi-Newton methods on convex problems (excepting DFP method). Intensive numerical experiments on minimizing functions with different dimensions and complexities showed that the BFGS method may require a large number of iterations or function and gradient evaluations on certain problems [24]. The sources of inefficiency of the BFGS method may be caused by a poor initial approximation to the Hessian or, more important, by the ill-conditioning of the Hessian approximations along the iterations, thus leading to a poorly defined search direction.

Two important tools in the analysis of the properties and of the convergence of the BFGS method are the *trace* and the *determinant* of the standard  $B_{k+1}$  given by (1.6). The trace of a matrix is exactly the sum of its eigenvalues. The determinant of a matrix is the product of its eigenvalues. By direct computation from (1.6) we get:

$$\text{tr}(B_{k+1}) = \text{tr}(B_k) - \frac{\|B_k s_k\|^2}{s_k^T B_k s_k} + \frac{\|y_k\|^2}{y_k^T s_k}. \quad (1.9)$$

On the other hand

$$\begin{aligned} \det(B_{k+1}) &= \det \left[ B_k \left( I - \frac{s_k s_k^T B_k}{s_k^T B_k s_k} + \frac{B_k^{-1} y_k y_k^T}{y_k^T s_k} \right) \right] \\ &= \det(B_k) \det \left( I - s_k \frac{(B_k s_k)^T}{s_k^T B_k s_k} + B_k^{-1} y_k \frac{y_k^T}{y_k^T s_k} \right). \end{aligned}$$

Now, applying the identity (see [39] and [48])

$$\det(I + u_1 u_2^T + u_3 u_4^T) = (1 + u_1^T u_2)(1 + u_3^T u_4) - (u_1^T u_4)(u_2^T u_3) \quad (1.10)$$

where

$$u_1 = -s_k, \quad u_2 = \frac{B_k s_k}{s_k^T B_k s_k}, \quad u_3 = B_k^{-1} y_k \quad \text{and} \quad u_4 = \frac{y_k}{y_k^T s_k},$$

we obtain:

$$\det(B_{k+1}) = \det(B_k) \frac{y_k^T s_k}{s_k^T B_k s_k}. \quad (1.11)$$

From the right hand side of (1.6) observe that the Hessian approximation  $B_k$  is modified by two correction matrices, each having rank one. As we know, the efficiency of the BFGS method is dependent on the structure of the eigenvalues of the approximation to the Hessian matrix (see Nocedal [35]). Powell [43] and Byrd, Liu and Nocedal [13] showed that the BFGS method actually suffers more from the large eigenvalues than from the small ones. Analyzing  $\text{tr}(B_{k+1})$  observe that the second term of the right hand side of (1.9) is negative. Therefore, it produces a shift of the eigenvalues of  $B_{k+1}$  to the left. Thus, the BFGS method is able to correct large eigenvalues. On the other hand, the third term of the right hand side of (1.9) being positive produces a shift of the eigenvalues of  $B_{k+1}$  to the right. If this term is large,  $B_{k+1}$  may have large eigenvalues, too. Therefore, a correction of the eigenvalues of  $B_{k+1}$  can be achieved by scaling the corresponding terms in (1.6) and this is the main motivation for which we use the scaled BFGS method. We emphasize that it must be a balance between these scaling of the corresponding terms in (1.6) (i.e. the eigenvalue shifting), otherwise the Hessian approximation could either approach singularity or become arbitrarily large, thus determining the failure of the method [35].

For strongly convex functions Nocedal [35] proved that the second term of the right hand side of (1.9) depends on  $\cos \theta_k$ , the angle between the steepest descent direction and the search direction, i.e. this term may be bounded as:

$$\frac{\|B_k s_k\|^2}{s_k^T B_k s_k} \geq \frac{\alpha_k}{c \cos^2 \theta_k}, \quad (1.12)$$

where  $c > 0$  is a constant. Therefore, this term tends to decrease the trace of  $B_{k+1}$  with a quantity which is proportional to  $\alpha_k / \cos^2 \theta_k$ . At the same time, Nocedal [35] proved that the third term of the right hand side of (1.9) is upper bounded, i.e.

$$\frac{\|y_k\|^2}{y_k^T s_k} \leq M, \quad (1.13)$$

where  $M > 0$  is a constant. Therefore, on strongly convex problems, the term that tends to increase the trace of  $B_{k+1}$  is bounded above for all  $k$ . With these, from (1.9) we obtain:

$$\text{tr}(B_{k+1}) \leq \text{tr}(B_k) - \frac{\alpha_k}{c \cos^2 \theta_k} + M. \quad (1.14)$$

The analysis includes two cases. Suppose that the stepsizes  $\alpha_k$  are bounded below. Observe that if  $\cos \theta_k$  is not very small, then some eigenvalues of  $B_{k+1}$  could become large because the second term of the right hand side of (1.14) could be significantly smaller than  $M$ . On the other hand, if  $\cos \theta_k$  is small, then the self correcting mechanism takes place, i.e. the middle term of the right hand side on (1.14) will be larger than  $M$ , thus decreasing the trace. Therefore, the smaller  $\cos \theta_k$  is, the faster the reduction in the trace of  $B_{k+1}$  (see Nocedal [35]). Now, suppose that the stepsizes  $\alpha_k$  tend to zero. In this case, Byrd, Nocedal and Yuan [14] proved that this is due to the

existence of very small eigenvalues in  $B_{k+1}$ , which cannot be monitored by means of the trace. But, BFGS updating formula has a strong self-correcting property subject to determinant of  $B_{k+1}$ , which can be used to show that, in fact,  $\alpha_k$  is bounded away from zero in mean. From (1.11) we see that when  $s_k^T B_k s_k$  is small relative to  $y_k^T s_k$ , then the determinant increases, showing that the small curvature of the model of the minimizing function  $f$  is corrected, thus increasing some eigenvalues. As a conclusion of this short presentation we see that the trace of  $B_{k+1}$  shows that for strongly convex functions, the eigenvalues of  $B_{k+1}$  cannot become too large, and the determinant of  $B_{k+1}$  shows that they cannot become too small [35]. Therefore, by *scaling the terms of the BFGS update (1.6) the structure of the eigenvalues of the  $B_{k+1}$  matrix is modified in such a way that the efficiency of the BFGS method could be improved*. This is the main motivation of scaling in BFGS update.

In order to improve the performances of the BFGS method, the *self-scaling BFGS methods* have been derived, firstly suggested and analyzed for minimization of the quadratic functions. Oren and Luenberger [38] scaled the Hessian approximation  $B_k$  before updating it, i.e. they replaced  $B_k$  by  $\tau_k B_k$ , where  $\tau_k$  is a self-scaling factor computed to reduce the condition number of  $R_k$  when it is applied to a quadratic function with Hessian  $G$ , where  $R_k = G^{1/2} H_k G^{1/2}$  and  $H_k$  is the current inverse approximation to the Hessian.

An extension of this self-scaling BFGS method was considered by Al-Baali [1], who introduced a simple modification:  $\tau_k = \min\{1, \tau_k\}$ . The numerical experiments in [1] showed that the modified self-scaling BFGS method is competitive versus the unscaled BFGS method. In the same line of efforts, Al-Baali [2] introduced a restricted class of self-scaling quasi-Newton methods which imposed some conditions on the Broyden family parameter and on the self-scaling factor  $\tau_k$ . The global convergence and the local superlinear convergence of this class of self-scaling methods with inexact line search were given by Al-Baali [2]. The numerical experiments with this restricted class of self-scaling quasi-Newton methods were reported by Al-Baali [3] on a set of small test unconstrained optimization problems up to 20 variables.

Another self-scaled BFGS method was proposed by Nocedal and Yuan [36] where the first two terms of the BFGS updating formula are scaled by the same factor  $y_k^T s_k / s_k^T B_k s_k$ , where  $s_k = x_{k+1} - x_k$  and  $y_k = g_{k+1} - g_k$ . They proved that this scaled BFGS method under inexact line search is globally convergent on general convex functions. They reported disappointing numerical results with their self-scaling BFGS method, this being consistent with the analysis given by Shanno and Phua [46].

This idea of scaling is now commonly applied only after the first iteration of a quasi-Newton method. A different approach was proposed by Powell [43] and further developed by Lalee and Nocedal [28] and Siegel [47]. Powell's idea was to work with a factorization  $H_k = Z_k Z_k^T$  of the inverse Hessian. On the other hand, Lalee and Nocedal [28] extended Powell's idea to scale down the columns of  $Z_k$  that are too large, as well as to scale up those which are too small. Siegel [47] suggested scaling up the last  $l$  columns of  $Z_k$ , where  $l$  is an integer parameter.

Many other modified BFGS methods were suggested. Using different function interpolation conditions, Biggs [9, 10] and Yuan [50] obtained some modified BFGS methods and proved their global convergence. The idea of their methods was to scale the third term of the BFGS updating formula. The modified BFGS method by Yuan used both gradient and function values information in one step.

Another self-scaling modified BFGS method was suggested by Aiping Liao [30]. In this method two positive scaling parameters which scale the second and the third terms of the BFGS updating formula were introduced, which correct the eigenvalues of  $B_k$  better than the original unscaled BFGS does. Numerical experiments support this claim and indicate that this scaled BFGS method may be competitive versus the standard unscaled BFGS method. The values of these parameters are computed in an adaptive way subject to a positive parameter. The global convergence of this two parameters scaled BFGS modified method is proved by using a tool introduced by Byrd and Nocedal [15].

A recent spectral scaling BFGS method was proposed by Cheng and Li [16]. In their method, the standard BFGS update is modified by introducing a positive scale factor  $\gamma_k$  to the third term of the BFGS updating formula, which is exactly the Barzilai and Borwein [8] parameter obtained by minimizing  $\|s_k - \gamma_k y_k\|^2$ . Comparisons of this spectral scaled BFGS method versus some other scaled modified BFGS methods given by Yuan [50], Al-Baali [3], Zhang and Xu [51] proved that this spectral scaled BFGS method is clearly more efficient and more robust.

Another very recent adaptive scaled BFGS method has been suggested by Andrei [6]. In this method the third term in the standard BFGS update formula is scaled by a positive factor in order to reduce the large eigenvalues of the approximation to the Hessian of the minimizing function. Under the inexact Wolfe line search, the global convergence of this adaptive scaled BFGS method is proved in very general conditions without assuming the convexity of the minimizing function. Intensive numerical experiments on unconstrained optimization test functions with a medium number of variables (up to 100) show that this variant of the scaled BFGS method is more efficient than the standard BFGS update or than some other well established scaled BFGS methods, including those of Biggs [9, 10], Cheng and Li [16] and Yuan [50].

In this paper we introduce a new scaled BFGS method with two parameters. The idea of this new two parameter scaled BFGS method is to improve its self-correcting property by scaling the first two terms of the standard BFGS update with a positive parameter and the third one with another positive parameter. In Section 2 we present the BFGS update with two scaling parameters and the corresponding DSBFGS algorithm. A number of 5 methods for selection of the scaling parameters in BFGS method, known in literature, are discussed in Section 3. These methods based in interpolation conditions [9, 10], spectral scaling [16], diagonal preconditioned and minimization the conjugacy conditions [6], scaling the last terms of the BFGS update with two parameters [30] or scaling the first two terms of the BFGS update with a parameter [36] are used to compare their computational performances versus the suggested BFGS update with two parameters. Section 4 presents an analysis of the proposed scaled BFGS update with two parameters. The parameter scaling the first two terms of the standard BFGS update is determined to cluster the eigenvalues of this matrix. The parameter scaling the third term is determined to reduce its large eigenvalues, thus obtaining a better distribution of them. The convergence analysis of the scaled BFGS update with two parameters is presented in Section 5. The analysis is based on the developments given in [6, 15 and 29]. We find that the scaled BFGS algorithm with two parameters is globally convergent in very general conditions without the convexity assumption of the minimizing function and when the scaling parameters are bounded. Our analysis is based on the trace of the BFGS approximation of the Hessian. In Section 6 some numerical results of the suggested scaled BFGS update with two parameters are given by using 80 unconstrained optimization test problems of medium dimensions (100 variables). Comparisons versus standard BFGS algorithm, as well as versus some other scaled BFGS algorithms by Biggs [9, 10], Cheng and Li [16], Yuan [50], Nocedal and Yuan [36] and Liao [30] are given. We have the computational evidence that our scaled BFGS algorithm with two parameters is more efficient than all these scaled BFGS algorithms. However, the adaptive scaled BFGS method by Andrei

[6], in which only the third term of the standard BFGS update is scaled with a positive factor, is much more efficient than the scaled BFGS method with two parameters suggested in this paper.

## 2. A scaled BFGS method with two parameters

Motivated by the idea of changing the structure of the eigenvalues of the BFGS approximation to the Hessian matrix by scaling its terms, in this paper we propose a *scaled BFGS method with two parameters* in which the updating of the approximation Hessian matrix  $B_{k+1}$  is computed as:

$$B_{k+1} = \delta_k \left[ B_k - \frac{B_k s_k s_k^T B_k}{s_k^T B_k s_k} \right] + \gamma_k \frac{y_k y_k^T}{y_k^T s_k}, \quad (2.1)$$

where  $\delta_k$  and  $\gamma_k$  are positive parameters. In our scaled BFGS method the parameter  $\delta_k$  is selected to cluster the eigenvalues of  $B_{k+1}$ . On the other hand,  $\gamma_k$  is determined to reduce the large eigenvalues of  $B_{k+1}$ , thus hoping to obtain a better distribution of the eigenvalues. It is worth saying that a variant of this scaled BFGS update was firstly considered by Oren and Luenberger [38] and further analyzed by Nocedal and Yuan [36], where  $\delta_k = y_k^T s_k / s_k^T B_k s_k$  and  $\gamma_k = 1$ . With  $\gamma_k = 1$ , other choices for  $\delta_k$  are given in Luenberger [33] and Oren [37], as well as in the references therein. If  $\delta_k = 1$  and  $\gamma_k = 1$ , then this method reduces to the standard BFGS method. Using the rank-one Sherman-Morrison-Woodbury update formula twice, from (2.1) we get  $H_{k+1} = B_{k+1}^{-1}$ , where

$$H_{k+1} = \frac{1}{\delta_k} \left[ H_k - \frac{H_k y_k s_k^T + s_k y_k^T H_k}{y_k^T s_k} + \left( \frac{\delta_k}{\gamma_k} + \frac{y_k^T H_k y_k}{y_k^T s_k} \right) \frac{s_k s_k^T}{y_k^T s_k} \right], \quad (2.2)$$

is the approximation to the inverse Hessian.

**Proposition 2.1.** *If the stepsize  $\alpha_k$  is determined by the Wolfe line search (1.4) and (1.5),  $B_k$  is positive definite and  $\gamma_k > 0$ , then  $B_{k+1}$  given by (2.1) is also positive definite.*

*Proof* Using the symmetry and positivity of  $B_k$ , we have

$$(s_k^T B_k z)^2 \leq (s_k^T B_k s_k)(z^T B_k z),$$

with equality if  $z = 0$  or  $s_k = 0$ . On the other hand, by the Wolfe line search (1.4) and (1.5) we have that  $y_k^T s_k > 0$ . Therefore, using the above inequality we get:

$$\begin{aligned} z^T B_{k+1} z &= \delta_k z^T B_k z - \delta_k \frac{z^T B_k s_k s_k^T B_k z}{s_k^T B_k s_k} + \gamma_k \frac{z^T y_k y_k^T z}{y_k^T s_k} \\ &= \delta_k z^T B_k z - \delta_k \frac{(z^T B_k s_k)^2}{s_k^T B_k s_k} + \gamma_k \frac{(z^T y_k)^2}{y_k^T s_k} \geq \gamma_k \frac{(z^T y_k)^2}{y_k^T s_k} > 0, \end{aligned}$$

for any nonzero  $z$ . ■

The above proposition says that  $B_{k+1}$  given by (2.1) with  $\gamma_k > 0$  inherits the positive definiteness of  $B_k$  and it does not rely on the line search used or on the convexity of the function  $f$ . Moreover, observe that this property is not dependent on the values of the parameter  $\delta_k$ .

Therefore, (2.1) is well defined if  $y_k^T s_k > 0$ , which is satisfied if the stepsize is determined by the Wolfe line search conditions (1.4) and (1.5). The corresponding scaled BFGS algorithm with two parameters can be presented as follows.

---

**Double Parameter Scaled BFGS algorithm – DSBFGS**

---

1. Initialization. Choose an initial point  $x_0 \in R^n$  and an initial positive definite matrix  $H_0$ . (Usually,  $H_0 = I$ .) Choose the constants  $\sigma$ ,  $\rho$  with  $0 < \sigma < \rho < 1$ , and  $\varepsilon > 0$  sufficiently small. Compute  $g_0 = \nabla f(x_0)$ . Set  $d_0 = -g_0$ . Set  $k = 0$ .
  2. Test a criterion for stopping the iterations. For example, if  $\|g_k\| \leq \varepsilon$ , then stop the iterations. Otherwise, continue with step 3.
  3. Compute the stepsize  $\alpha_k > 0$  satisfying the Wolfe line search conditions (1.4) and (1.5).
  4. Compute  $x_{k+1} = x_k + \alpha_k d_k$ ,  $f_{k+1} = f(x_{k+1})$  and  $g_{k+1} = \nabla f(x_{k+1})$ . Set  $s_k = x_{k+1} - x_k$ ,  $y_k = g_{k+1} - g_k$ .
  5. Compute the scaling factors  $\delta_k$  and  $\gamma_k$ .
  6. Update the inverse Hessian  $H_k$  using (2.2).
  7. Compute the search direction as  $d_{k+1} = -H_{k+1} g_{k+1}$ .
  8. Set  $k = k + 1$  and continue with step 2. ■
- 

Observe that if  $\delta_k = 1$  and  $\gamma_k = 1$  for all  $k = 0, 1, \dots$ , then the above algorithm is exactly the standard BFGS algorithm. For different values of the parameters  $\delta_k$  and  $\gamma_k$  in (2.1) (or (2.2)), different scaled BFGS algorithms are obtained. The algorithm is very easy to implement, but it is applicable only for solving small and medium unconstrained optimization problems.

### 3. Selection of the scaling parameters in BFGS method

To implement the DSBFGS algorithm, some procedures for  $\delta_k$  and  $\gamma_k$  in step 5 must be given. For the scaling parameter  $\gamma_k$  in (2.1) some values have been proposed in literature, as follows.

1) *Scaled BFGS with different interpolation conditions* (Biggs [9, 10] and Yuan [50]). Observe that the quasi-Newton step  $d_k = -H_k g_k$  is a stationary point of the following problem:

$$\min_{d \in R^n} \phi_k(d) = f(x_k) + g_k^T d + \frac{1}{2} d^T B_k d. \quad (3.1)$$

Since for small  $d$ ,  $\phi_k(d) \approx f(x_k + d)$ , it follows that the problem (3.1) is an approximation to the problem (1.1) near the current point  $x_k$ . From (3.1) we have that

$$\phi_k(0) = f(x_k), \quad \nabla \phi_k(0) = g(x_k), \quad (3.2)$$

and the quasi-Newton condition  $H_k y_{k-1} = s_{k-1}$  is equivalent to

$$\nabla \phi_k(x_{k-1} - x_k) = g(x_{k-1}). \quad (3.3)$$

Therefore  $\phi_k(x - x_k)$  is a quadratic interpolation of  $f(x)$  at  $x_k$  satisfying the above conditions (3.2) and (3.3).

If the objective function is cubic along the line segment connecting  $x_{k-1}$  and  $x_k$  and the Hermite interpolation is used on the same line between  $x_{k-1}$  and  $x_k$ , then the following condition holds

$$s_{k-1}^T \nabla^2 f(x_k) s_{k-1} = 4s_{k-1}^T g_k + 2s_{k-1}^T g_{k+1} - 6(f(x_{k-1}) - f(x_k)). \quad (3.4)$$



Biggs [9, 10] considers the update (2.1) with  $\delta_k = 1$  and the value of  $\gamma_k$  chosen in such a way that the new approximate Hessian satisfies the reasonable condition

$$s_{k-1}^T B_k s_{k-1} = 4s_{k-1}^T g_k + 2s_{k-1}^T g_{k+1} - 6(f(x_{k-1}) - f(x_k)). \quad (3.5)$$

Therefore, the value of  $\gamma_k$  proposed by Biggs is

$$\gamma_k = \frac{6}{y_k^T s_k} (f(x_k) - f(x_{k+1}) + s_k^T g_{k+1}) - 2. \quad (3.6)$$

For one-dimensional problems, Wang and Yuan [49] showed that the scaled BFGS (2.1) with  $\delta_k = 1$  and  $\gamma_k$  given by (3.6) and without line search is R-linear convergent.

In the same line of developments, Yuan [50] considered that the approximate function  $\phi_k(d)$  satisfies the interpolation condition

$$\phi_k(x_{k-1} - x_k) = f(x_{k-1}) \quad (3.7)$$

instead of (3.3) and determines the following value for the scaling parameter

$$\gamma_k = \frac{2}{y_k^T s_k} (f(x_k) - f(x_{k+1}) + s_k^T g_{k+1}). \quad (3.8)$$

For uniformly convex functions it is easy to prove that there exists a constant  $\xi > 0$  such that for all  $k$ ,  $\gamma_k \in [\xi, 2]$ . Powell [42] showed that the scaled BFGS method (2.1) with  $\delta_k = 1$  and  $\gamma_k$  given by (3.8) is globally convergent for convex functions with inexact line search. However, for general nonlinear functions, the inexact line search does not involve the positivity of  $\gamma_k$ . In these cases Yuan restricted  $\gamma_k$  in the interval  $[0.01, 100]$  and proved the global convergence of this variant of the scaled BFGS method.

2) *Spectral scaled BFGS* (Cheng and Li [16]). In this method, introduced by Cheng and Li [16], in (2.1),  $\delta_k = 1$  and the scaling parameter  $\gamma_k$  is computed as

$$\gamma_k = \frac{y_k^T s_k}{\|y_k\|^2}, \quad (3.9)$$

obtained as solution of the problem:  $\min \|s_k - \gamma_k y_k\|^2$ . Observe that (3.9) is exactly one of the spectral stepsizes introduced by Barzilai and Borwein [8]. Therefore, the scaled BFGS method given by (2.1) with  $\delta_k = 1$  and  $\gamma_k$  given by (3.9) is viewed as the spectral scaled BFGS method. Under classical assumptions it is proved that this spectral scaled BFGS method with Wolfe line search is globally convergent and R-linear convergent for convex optimization problems. Using some test problems with dimensions between 10 and 500 from the CUTE collection [11], Cheng and Li [16] present the computational evidence that their spectral scaled BFGS algorithm is top performer versus the standard BFGS and the scaled BFGS algorithms by Al-Baali [3], Yuan [50] and Zhang and Xu [51].

3) *Scaled BFGS with diagonal preconditioning and conjugacy condition* (Andrei [6]). In [6] Andrei introduced another scaled BFGS update given by (2.1), in which  $\delta_k = 1$  and the scaling parameter  $\gamma_k$  is computed in an adaptive manner as:

$$\gamma_k = \min \left\{ \frac{y_k^T s_k}{\|y_k\|^2 + \beta_k}, 1 \right\}, \quad (3.10)$$

where  $\beta_k > 0$  for all  $k = 0, 1, \dots$ . Since under the Wolfe line search conditions (1.4) and (1.5)  $y_k^T s_k > 0$  for all  $k = 0, 1, \dots$ , it follows that  $\gamma_k$  given by (3.10) is bounded away from zero, i.e.

$0 < \gamma_k \leq 1$ . Therefore, the motivation behind this adaptive computation of the scaling parameter  $\gamma_k$  as in (3.10) is that if  $\gamma_k \leq 1$ , then the structure of the large eigenvalues of  $B_{k+1}$  is corrected by their shifting to the left. It is proved that if  $\gamma_k$  is selected as in (3.10), where  $\beta_k > 0$  for all  $k = 0, 1, \dots$ , then the large eigenvalues of  $B_{k+1}$  given by (2.1) with  $\delta_k = 1$  are shifted to the left [6]. In [6], intensive numerical experiments showed that this scaled BFGS algorithm with  $\beta_k = |s_k^T g_{k+1}|$  is the best one, being more efficient and more robust versus the standard BFGS algorithm as well as versus some other scaled BFGS algorithms, including the versions of Biggs [9, 10], Yuan [50] and Cheng and Li [16]. The theoretical justification of this selection of the parameter  $\gamma_k$  is as follows. To have a good algorithm, we hope that  $\gamma_k I$  is a diagonal preconditioner of  $\nabla^2 f(x_{k+1})$  that reduces the condition number to the inverse of  $\nabla^2 f(x_{k+1})$ , i.e. it reduces the large eigenvalues. Such matrix  $\gamma_k I$  should be a rough approximation to the inverse of  $\nabla^2 f(x_{k+1})$ . Therefore,  $\gamma_k$  can be computed to minimize  $\|s_k - \gamma_k y_k\|^2$ . On the other hand, for nonlinear functions, the classical conjugacy condition used by Hestenes and Stiefel [27] for quadratic functions which incorporate the second-order information is  $d_{k+1}^T y_k = -s_k^T g_{k+1}$ . Therefore, in our scaled BFGS algorithm we want  $\gamma_k I$  to be a diagonal preconditioner of  $\nabla^2 f(x_{k+1})$  and also to minimize the conjugacy condition, i.e.  $\gamma_k$  can be selected to minimize a combination of these two conditions:

$$\min\{\|s_k - \gamma_k y_k\|^2 + \gamma_k^2 |s_k^T g_{k+1}|\}.$$

Therefore, DSBFGS algorithm may be particularized with  $\gamma_k$  selected as in (3.10), where  $\beta_k = |s_k^T g_{k+1}|$ , and  $\delta_k = 1$  (see [6]).

4) *Scaling the last terms of the BFGS update with two parameters* (Liao [30]). In another avenue of research Liao [30] introduced the modified (scaled) BFGS method with two parameters:

$$B_{k+1} = B_k - \delta_k \frac{B_k s_k s_k^T B_k}{s_k^T B_k s_k} + \gamma_k \frac{y_k y_k^T}{y_k^T s_k} \quad (3.11)$$

and proved that this scaled BFGS method with two positive parameters  $\delta_k$  and  $\gamma_k$  corrects the large eigenvalues better than the standard BFGS method given by (1.6) does. In other words, it is proved that this scaled BFGS method has a strong self-correcting property with respect to the determinant [30]. In Liao's method, the parameters scaling the terms in the BFGS update are computed in an adaptive way subject to the values of a positive parameter as:

$$(\delta_k, \gamma_k) = \begin{cases} \left( \frac{s_k^T B_k s_k}{s_k^T B_k s_k + y_k^T s_k}, \frac{y_k^T s_k}{s_k^T B_k s_k + y_k^T s_k} \right), & \text{if } \frac{s_k^T B_k s_k}{s_k^T B_k s_k + y_k^T s_k} \geq \tau_k, \\ (\tau_k, 1), & \text{otherwise,} \end{cases} \quad (3.12)$$

where  $0 < \tau_k < 1$ . Liao [30] proposed  $\tau_k = \exp(-1/k^2)$ . Using a tool given by Byrd and Nocedal [15], Liao proved that the scaled BFGS method given by (3.11)-(3.12) with the Wolfe line search generates iterates which converge superlinearly to the optimal solution. Limited numerical experiments with Liao's scaled BFGS method proved that it is competitive with the standard BFGS method and it corrects large eigenvalues better than the standard BFGS method.

5) *Scaling the first two terms of the BFGS update with a parameter* (Oren and Luenberger [38] and Nocedal and Yuan [36]). Now, concerning the selection of  $\delta_k$  in (2.1) with  $\gamma_k = 1$ , Oren and Luenberger [38] suggested  $\delta_k = y_k^T s_k / s_k^T B_k s_k$  being one of the best as it simplifies the analysis. Furthermore, Nocedal and Yuan [36] presented a deep analysis of this scaling quasi-Newton method and showed that even if the corresponding algorithm with inexact line search is superlinear convergent on general functions, it is computationally expensive as regards the steplength computation. In other words, the numerical results with this scaling BFGS algorithm are not convincing. In this paper we propose another strategy for selecting the scaling parameter  $\delta_k$ .

#### 4. Analysis of the scaled BFGS update with two parameters

As we know, the performances of the BFGS method are much improved if the eigenvalues of the iteration matrix (2.1) are clustered (see [5]). From (2.1) observe that

$$tr(B_{k+1}) = \delta_k tr(B_k) - \delta_k \frac{\|B_k s_k\|^2}{s_k^T B_k s_k} + \gamma_k \frac{\|y_k\|^2}{y_k^T s_k}. \quad (4.1)$$

Nocedal [35] proved that the third term on the right hand side of (1.9) is bounded by a positive constant as in (1.13). In our algorithm the third term of (4.1) is reduced by the selection of  $\gamma_k$  as in (3.10). Since the trace of a matrix is the sum of its eigenvalues, in our scaled BFGS update with two parameters (2.1) we suggest the parameter  $\delta_k$  should be selected in such a way that  $tr(B_{k+1})$  given by (4.1) to be equal to  $n$ . The idea is to select  $\delta_k$  such that the eigenvalues of  $B_{k+1}$  to be clustered. Therefore, from the equation  $tr(B_{k+1}) = n$ , where  $\gamma_k$  is given by (3.10) we obtain:

$$\delta_k = \frac{n - \gamma_k \frac{\|y_k\|^2}{y_k^T s_k}}{n - \frac{\|B_k s_k\|^2}{s_k^T B_k s_k}}. \quad (4.2)$$

A characterization of  $\delta_k$  in (4.2) is as follows.

**Proposition 4.1.** *Let  $\delta_k$  be computed as in (4.2). Then, for any  $k = 0, 1, \dots$ ,  $\delta_k$  is positive and close to 1.*

*Proof* Observe that along the iterations  $|s_k^T g_{k+1}| \rightarrow 0$ . Therefore,  $\|y_k\|^2 / (\|y_k\|^2 + |s_k^T g_{k+1}|)$  is close to 1. On the other hand,  $B_k$  is symmetric and positive definite. Therefore, it has real and positive eigenvalues:  $\lambda_1, \dots, \lambda_n$ . Since  $B_k$  is nonsingular and  $tr(B_k) = n$ , it follows that for any  $i = 1, \dots, n$ ,  $\lambda_i > 0$  such that  $\sum_{i=1}^n \lambda_i = n$ . Observe that  $\|B_0 s_0\|^2 = s_0^T B_0 s_0$ . But, for  $k$  sufficiently large,  $0 < \|B_k s_k\|^2 < 1$  and  $0 < s_k^T B_k s_k < 1$ . Since  $\|B_k s_k\|^2$  and  $s_k^T B_k s_k$  are approximately of the same order of magnitude, it follows that  $n \gg \|B_k s_k\|^2 / s_k^T B_k s_k$ . Therefore,

we have  $n \gg \gamma_k \|y_k\|^2 / y_k^T s_k$  and  $n \gg \|B_k s_k\|^2 / s_k^T B_k s_k$ , i.e. for any  $k=0,1,\dots$ ,  $\delta_k$  is positive and close to 1. Observe that the bigger  $n$  is, the closer to 1  $\delta_k$  is. ■

In order to investigate the properties and the rate of convergence of the DSBFGS algorithm let us consider the analysis on minimization of the strictly convex quadratic function

$$f(x) = \frac{1}{2}(x - x^*)^T G(x - x^*) + f(x^*) \quad (4.3)$$

using the Newton method  $x_{k+1} = x_k - \alpha_k H_k g_k$ , where  $H_k$  is a positive definite matrix and  $\alpha_k$  is a stepsize. When the Newton method with the exact line search is applied to minimize (4.3), then the single-step convergence rate can be expressed as:

$$f(x_{k+1}) - f(x^*) \leq \left[ \frac{\kappa(R_k) - 1}{\kappa(R_k) + 1} \right]^2 (f(x_k) - f(x^*)),$$

where  $R_k = G^{1/2} H_k G^{1/2}$  and  $\kappa(R_k)$  is the condition number of  $R_k$ , i.e. the ratio of the largest to the smallest eigenvalues of  $R_k$ . Observe that for the steepest descent method  $H_k = I$ , so  $R_k = G$  and the single-step convergence rate is linear, with a rate bounded in term of  $\kappa(G)$ . Luenberger [32] proved that the quasi-Newton DFP method with exact line search applied to minimize (4.3) might cause  $\kappa(R_k) > \kappa(G)$  at some iterations. Therefore, in some cases DFP method may be inferior to the steepest descent method. Dixon [20] showed that the Broyden class of the quasi-Newton methods with exact line search produces the same iterations for general functions. Therefore, in some cases, the BFGS method with exact line search may be inferior to the steepest descent method (see [16]). The following theorem shows that the algorithm DSBFGS can avoid such cases. For this we need to introduce the following result of Loewner [31].

**Proposition 4.2.** *Let  $A \in R^{n \times n}$  be a symmetric matrix with eigenvalues  $\lambda_n \leq \lambda_{n-1} \leq \dots \leq \lambda_1$  and let  $a \in R^n$  be an arbitrary nonzero vector. Denote the eigenvalues of the matrix  $\bar{A} = A - aa^T$  by  $\mu_n \leq \mu_{n-1} \leq \dots \leq \mu_1$ . Then, we have  $\mu_n \leq \lambda_n \leq \mu_{n-1} \leq \lambda_{n-1} \leq \dots \leq \mu_1 \leq \lambda_1$ . ■*

**Theorem 4.1.** *If we apply the algorithm DSBFGS with  $\gamma_k$  selected as in (3.10), where  $\beta_k \geq 0$ , with exact line search and  $B_0 = I$  to minimize (4.3), then  $\kappa(R_k) \leq \kappa(G)$ , where  $R_k = G^{1/2} H_k G^{1/2}$  and  $H_k = B_k^{-1}$ .*

*Proof* The proof is given by induction as in [16] (see also [38]). Define  $r_k = G^{1/2} s_k$ . Observe that  $R_k$  is similar to  $H_k G$ . For  $k=0$  the conclusion of the theorem is clear since  $H_0 = I$ . Suppose that for some  $k \geq 0$ ,  $\kappa(R_k) \leq \kappa(G)$ . Now, let us write (2.1) as

$$H_{k+1}^{-1} = \delta_k H_k^{-1} - \delta_k \frac{H_k^{-1} s_k s_k^T H_k^{-1}}{s_k^T H_k^{-1} s_k} + \gamma_k \frac{y_k y_k^T}{y_k^T s_k}. \quad (4.4)$$

Now, pre-multiplying and post-multiplying both sides of the above equality by  $G^{-1/2}$  and using the relation  $y_k = G s_k$  we get:

$$R_{k+1}^{-1} = \delta_k R_k^{-1} - \delta_k \frac{R_k^{-1} r_k r_k^T R_k^{-1}}{r_k^T R_k^{-1} r_k} + \gamma_k \frac{r_k r_k^T}{r_k^T r_k}. \quad (4.5)$$

Let the eigenvalues of  $R_k^{-1}$  be arranged as  $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_n > 0$ . Define the matrix:

$$P = \delta_k \left[ R_k^{-1} - \frac{R_k^{-1} r_k r_k^T R_k^{-1}}{r_k^T R_k^{-1} r_k} \right]. \quad (4.6)$$

Observe that  $P r_k = 0$ . Therefore, the matrix  $P$  has zero as its eigenvalue, which corresponds to  $r_k$  as eigenvector. Observe that  $P$  in (4.6) can be written as:

$$P = \delta_k R_k^{-1} - \frac{\delta_k}{r_k^T R_k^{-1} r_k} (R_k^{-1} r_k)(R_k^{-1} r_k)^T. \quad (4.7)$$

Now, if we denote the eigenvalues of  $P$  by  $\mu_1 \geq \mu_2 \geq \dots \geq \mu_n = 0$ , and having in view the structure of  $P$  given by (4.7), then by Proposition 4.2 we have

$$\lambda_1 \geq \mu_1 \geq \lambda_2 \geq \mu_2 \geq \dots \geq \lambda_n \geq \mu_n = 0. \quad (4.8)$$

From (4.5) we have

$$R_{k+1}^{-1} = P + \gamma_k \frac{r_k r_k^T}{r_k^T r_k}.$$

Therefore, since  $P r_k = 0$  we have  $R_{k+1}^{-1} r_k = \gamma_k r_k$ , i.e.  $R_{k+1}^{-1}$  has  $\gamma_k$  as its eigenvalue which corresponds to  $r_k$  as eigenvector. Since  $P$  is symmetric and  $r_k$  is an eigenvector of  $P$ , it follows that every other eigenvector of  $P$  is orthogonal to  $r_k$ . Let us consider  $w_j$  as an eigenvector of  $P$  corresponding to the eigenvalue  $\mu_j$  for some  $j = 1, \dots, n-1$ . Then, we have  $R_{k+1}^{-1} w_j = P w_j = \mu_j w_j$ ,  $j = 1, \dots, n-1$ . Therefore,  $\mu_1, \mu_2, \dots, \mu_{n-1}, \gamma_k$  are eigenvalues of  $R_{k+1}^{-1}$ .

Since for any nonsingular matrix  $X$  we have that  $\kappa(X) = \kappa(X^{-1})$ , by inductive assumption it follows that

$$\kappa(R_k^{-1}) \leq \kappa(G^{-1}). \quad (4.9)$$

Now, let us consider that  $h_1$  and  $h_2$  are the largest and the smallest eigenvalues of  $G^{-1}$  respectively. Then, (4.9) implies that  $[\lambda_n, \lambda_1] \subseteq [h_2, h_1]$ . Therefore, from (4.8) it follows that  $\mu_1, \mu_2, \dots, \mu_{n-1} \in [h_2, h_1]$ .

On the other hand, observe that

$$\gamma_k = \frac{y_k^T s_k}{\|y_k\|^2 + \beta_k} \leq \frac{y_k^T s_k}{\|y_k\|^2}.$$

But, the Rayleigh quotient of  $G^{-1}$  is:

$$\frac{y_k^T s_k}{\|y_k\|^2} = \frac{s_k^T G s_k}{s_k^T G^2 s_k} = \frac{r_k^T r_k}{r_k^T G r_k}.$$

Therefore,  $\gamma_k$  is smaller than the Rayleigh quotient of  $G^{-1}$ . Thus,  $\gamma_k \in [h_2, h_1]$ . With this we have proved that all the eigenvalues of  $R_{k+1}^{-1}$  are in the interval  $[h_2, h_1]$ . Therefore,  $\kappa(R_{k+1}^{-1}) \leq \kappa(G^{-1})$ , i.e.

$$\kappa(R_{k+1}) \leq \kappa(G),$$

which completes the proof of the theorem. ■

From the proof of Theorem 4.1 we see that the parameter  $\gamma_k$  is the key parameter in the economy of the DSBFGS algorithm.  $\delta_k$  has only a marginal effect in this theorem. However, selected as in (4.2), the importance of the parameter  $\delta_k$  consists in clustering the eigenvalues of the iteration matrix.

### 5. Convergence analysis of the scaled BFGS update with two parameters

In this section we analyze the convergence of a variant of the scaled BFGS update with two parameters, i.e. of the DSBFGS algorithm where the scaling parameters  $\gamma_k$  and  $\delta_k$  are computed as in (3.10) and (4.2) with  $\beta_k = |s_k^T g_{k+1}|$ , respectively. Assume that the level set  $S = \{x : f(x) \leq f(x_0)\}$  is bounded. From the first Wolfe condition (1.4) it follows that the sequence  $\{f(x_k)\}$  is nonincreasing and therefore  $\lim_{k \rightarrow \infty} f(x_k)$  exists. Besides,  $x_k \in S$ . In order to establish the global convergence of the algorithm DSBFGS, some useful propositions are proved as follows, where  $\gamma_k$  is computed as in (3.10) and  $\delta_k$  is computed as in (4.2) with  $\beta_k = |s_k^T g_{k+1}|$ . Our analysis is based on the same principles as those presented by Andrei [6] (see also Li and Fukushima [29] and by Byrd and Nocedal [15]).

**Proposition 5.1.** *Let  $\delta_k$  be computed as in (4.2) for  $k = 0, 1, \dots$ . Then, there are the positive constants  $0 < \delta < \Delta$  such that for any  $j = 0, 1, \dots, k$ ,*

$$\delta < \delta_k \delta_{k-1} \cdots \delta_j < \Delta. \quad (5.1)$$

*Proof* From Proposition 4.1 it follows that  $\delta_k$  is close to 1 for any  $k = 0, 1, \dots$ . As a consequence, there are the positive constants  $0 < \delta < \Delta$  such that any product of the form  $\delta_k \delta_{k-1} \cdots \delta_j$ , for any  $j = 0, 1, \dots$ , is bounded as in (5.1).  $\blacksquare$

**Proposition 5.2.** *Consider the scaled  $B_{k+1}$  update with two parameters given by (2.1), where  $\gamma_k$  and  $\delta_k$  are computed as in (3.10) and (4.2), with  $\beta_k = |s_k^T g_{k+1}|$ , respectively. Then*

$$\text{tr}(B_{k+1}) \leq \Delta \text{tr}(B_0) + (\Delta k + 1) \quad (5.2)$$

and

$$\sum_{i=0}^k \frac{\|B_i s_i\|^2}{s_i^T B_i s_i} \leq \frac{\Delta}{\delta} (\text{tr}(B_0) + k) + \frac{1}{\delta}. \quad (5.3)$$

*Proof* Observe that

$$\begin{aligned} \text{tr}(B_{k+1}) &= \delta_k \text{tr}(B_k) - \delta_k \frac{\|B_k s_k\|^2}{s_k^T B_k s_k} + \gamma_k \frac{\|y_k\|^2}{y_k^T s_k} \\ &= \delta_k \left( \delta_{k-1} \text{tr}(B_{k-1}) - \delta_{k-1} \frac{\|B_{k-1} s_{k-1}\|^2}{s_{k-1}^T B_{k-1} s_{k-1}} + \gamma_{k-1} \frac{\|y_{k-1}\|^2}{y_{k-1}^T s_{k-1}} \right) - \delta_k \frac{\|B_k s_k\|^2}{s_k^T B_k s_k} + \gamma_k \frac{\|y_k\|^2}{y_k^T s_k} \\ &= \dots \\ &= \delta_k \delta_{k-1} \cdots \delta_0 \text{tr}(B_0) \\ &\quad - \delta_k \delta_{k-1} \cdots \delta_0 \frac{\|B_0 s_0\|^2}{s_0^T B_0 s_0} + \delta_k \delta_{k-1} \cdots \delta_1 \gamma_0 \frac{\|y_0\|^2}{y_0^T s_0} \\ &\quad - \delta_k \delta_{k-1} \cdots \delta_1 \frac{\|B_1 s_1\|^2}{s_1^T B_1 s_1} + \delta_k \delta_{k-1} \cdots \delta_2 \gamma_1 \frac{\|y_1\|^2}{y_1^T s_1} \\ &\quad \dots \end{aligned}$$

$$\begin{aligned}
& -\delta_k \delta_{k-1} \frac{\|B_{k-1} s_{k-1}\|^2}{s_{k-1}^T B_{k-1} s_{k-1}} + \delta_k \gamma_{k-1} \frac{\|y_{k-1}\|^2}{y_{k-1}^T s_{k-1}} \\
& -\delta_k \frac{\|B_k s_k\|^2}{s_k^T B_k s_k} + \gamma_k \frac{\|y_k\|^2}{y_k^T s_k}.
\end{aligned} \tag{5.4}$$

But, for any  $i = 0, \dots, k$ ,

$$\gamma_i \frac{\|y_i\|^2}{y_i^T s_i} = \frac{y_i^T s_i}{\|y_i\|^2 + |s_i^T g_{i+1}|} \frac{\|y_i\|^2}{y_i^T s_i} = \frac{\|y_i\|^2}{\|y_i\|^2 + |s_i^T g_{i+1}|} \leq 1.$$

Therefore, since by Proposition 5.1 there are the positive constants  $0 < \delta < \Delta$  such that for any  $j = 0, 1, \dots, k$ ,  $\delta < \delta_k \delta_{k-1} \dots \delta_j < \Delta$ , it follows that

$$tr(B_{k+1}) \leq \Delta tr(B_0) - \sum_{i=0}^k \delta \frac{\|B_i s_i\|^2}{s_i^T B_i s_i} + \sum_{k=1}^k \Delta + 1 \leq \Delta tr(B_0) + \Delta k + 1. \tag{5.5}$$

From (5.5) we get (5.2).

On the other hand, since  $B_{k+1}$  is positive definite,  $tr(B_{k+1}) > 0$ . Therefore (5.3) is true.  $\blacksquare$

**Remark 5.1** If  $B_0 = I$ , then

$$tr(B_{k+1}) \leq \Delta n + (\Delta k + 1) \quad \text{and} \quad \sum_{i=0}^k \frac{\|B_i s_i\|^2}{s_i^T B_i s_i} \leq \frac{\Delta}{\delta} (n + k) + \frac{1}{\delta}.$$

Observe that the last inequality in (5.5) shows that the largest eigenvalue of  $B_{k+1}$  is strictly smaller than  $\Delta tr(B_0) + (\Delta k + 1)$ . Therefore, the scaled DSBFGS method with  $\gamma_k$  given by (3.10) and  $\delta_k$  given by (4.2) with  $\beta_k = |s_k^T g_{k+1}|$  has a good self-correcting property subject to the trace, i.e. it may be more efficient than the standard BFGS in correcting the large eigenvalues.

**Proposition 5.3.** *If for all  $k$ ,  $\gamma_k \geq m$ , where  $m > 0$  is a constant, and  $\delta_k \geq \theta$ , where  $\theta > 0$  is a constant, then there is a constant  $c > 0$  such that for all  $k$  sufficiently large:*

$$\prod_{i=0}^k \alpha_i \geq c^k. \tag{5.6}$$

*Proof* Considering the identity (1.10), the determinant of the scaled  $B_{k+1}$  given by (2.1) is as follows:

$$\begin{aligned}
\det(B_{k+1}) &= \det \left( \delta_k B_k \left( I - \frac{s_k s_k^T B_k}{s_k^T B_k s_k} + \frac{\gamma_k}{\delta_k} \frac{B_k^{-1} y_k y_k^T}{y_k^T s_k} \right) \right) \\
&= \det(\delta_k B_k) \det \left( I - s_k \frac{(B_k s_k)^T}{s_k^T B_k s_k} + \frac{\gamma_k}{\delta_k} (B_k^{-1} y_k) \frac{y_k^T}{y_k^T s_k} \right) \\
&= \delta_k^n \det(B_k) \frac{\gamma_k}{\delta_k} \frac{y_k^T s_k}{s_k^T B_k s_k}.
\end{aligned} \tag{5.7}$$

Therefore,

$$\begin{aligned}
\det(B_{k+1}) &= \delta_k^{n-1} \gamma_k \frac{y_k^T s_k}{s_k^T B_k s_k} \det(B_k) \\
&= \left( \delta_k^{n-1} \gamma_k \frac{y_k^T s_k}{s_k^T B_k s_k} \right) \left( \delta_{k-1}^{n-1} \gamma_{k-1} \frac{y_{k-1}^T s_{k-1}}{s_{k-1}^T B_{k-1} s_{k-1}} \right) \det(B_{k-1}) \\
&= \left( \delta_k^{n-1} \gamma_k \frac{y_k^T s_k}{s_k^T B_k s_k} \right) \left( \delta_{k-1}^{n-1} \gamma_{k-1} \frac{y_{k-1}^T s_{k-1}}{s_{k-1}^T B_{k-1} s_{k-1}} \right) \cdots \left( \delta_0^{n-1} \gamma_0 \frac{y_0^T s_0}{s_0^T B_0 s_0} \right) \det(B_0) \\
&= \left( \prod_{i=0}^k \delta_i^{n-1} \gamma_i \frac{y_i^T s_i}{s_i^T B_i s_i} \right) \det(B_0). \tag{5.8}
\end{aligned}$$

But, for all  $i$ ,  $s_i^T B_i s_i \leq -\alpha_i s_i^T g_i$  and  $y_i^T s_i \geq -(1-\rho) s_i^T g_i$ . Besides, for all  $i$ ,  $\gamma_i \geq m$  and  $\delta_i \geq \theta$ . Therefore,

$$\det(B_{k+1}) \geq \det(B_0) \prod_{i=0}^k \theta^{n-1} m \frac{1-\rho}{\alpha_i} = \det(B_0) (\theta^{n-1})^{k+1} m^{k+1} (1-\rho)^{k+1} \prod_{i=0}^k \frac{1}{\alpha_i}. \tag{5.9}$$

Since  $\det(B_{k+1}) \leq \left( \frac{1}{n} \text{tr}(B_{k+1}) \right)^n$ , by using Proposition 5.2, we get

$$\det(B_{k+1}) \leq \left( \frac{1}{n} (\Delta \text{tr}(B_0) + \Delta k + 1) \right)^n.$$

Therefore,

$$\prod_{i=0}^k \alpha_i \geq \frac{\det(B_0) \theta^{(n-1)(k+1)} m^{k+1} (1-\rho)^{k+1}}{\det(B_{k+1})} \geq \frac{\det(B_0) \theta^{(n-1)(k+1)} m^{k+1} (1-\rho)^{k+1}}{\left( \frac{1}{n} (\Delta \text{tr}(B_0) + \Delta k + 1) \right)^n}. \tag{5.10}$$

When  $k$  is sufficiently large, (5.10) implies (5.6). ■

**Remark 5.2.** If  $B_0 = I$ , then

$$\prod_{i=0}^k \alpha_i \geq \frac{\theta^{(n-1)(k+1)} m^{k+1} (1-\rho)^{k+1}}{\left( \frac{1}{n} (\Delta n + \Delta k + 1) \right)^n}.$$

**Theorem 5.1.** Let  $\{x_k\}$  be generated by the algorithm DSBFGS with  $\gamma_k$  and  $\delta_k$  computed as in (3.10) and (4.2) with  $\beta_k = |s_k^T g_{k+1}|$ , respectively. Then

$$\liminf_{k \rightarrow \infty} \|g_k\| = 0. \tag{5.11}$$

*Proof* Assume that  $\|g_k\| > \Gamma > 0$ , for all  $k$ . Observe that  $B_k s_k = \alpha_k B_k d_k$ . Since  $f$  is bounded from below, from the first Wolfe condition (1.4) we have  $\sum_{k=0}^{\infty} (-s_k^T g_k) < \infty$ . Therefore,



$$\begin{aligned}
\infty &> \sum_{k=0}^{\infty} (-s_k^T g_k) = \sum_{k=0}^{\infty} \frac{1}{\alpha_k} s_k^T B_k s_k = \sum_{k=0}^{\infty} \frac{\|g_k\|}{\|B_k s_k\|} s_k^T B_k s_k \\
&= \sum_{k=0}^{\infty} \frac{s_k^T B_k s_k}{\|B_k s_k\|} \|g_k\| \frac{\|B_k s_k\|}{\|B_k s_k\|} = \sum_{k=0}^{\infty} \frac{s_k^T B_k s_k}{\|B_k s_k\|} \|g_k\| \frac{\alpha_k \|g_k\|}{\|B_k s_k\|} \\
&= \sum_{k=0}^{\infty} \|g_k\|^2 \alpha_k \frac{s_k^T B_k s_k}{\|B_k s_k\|^2} \geq \Gamma^2 \sum_{k=0}^{\infty} \alpha_k \frac{s_k^T B_k s_k}{\|B_k s_k\|^2}.
\end{aligned} \tag{5.12}$$

Now, from the geometric inequality, for any  $\Omega > 0$  there exists an integer  $k_0 > 0$  such that for any positive integer  $q$  we have:

$$q \left[ \prod_{k=k_0+1}^{k_0+q} \alpha_k \frac{s_k^T B_k s_k}{\|B_k s_k\|^2} \right]^{1/q} \leq \sum_{k=k_0+1}^{k_0+q} \alpha_k \frac{s_k^T B_k s_k}{\|B_k s_k\|^2} \leq \Omega. \tag{5.13}$$

Hence,

$$\begin{aligned}
\left[ \prod_{k=k_0+1}^{k_0+q} \alpha_k \right]^{1/q} &\leq \frac{\Omega}{q} \left[ \prod_{k=k_0+1}^{k_0+q} \frac{\|B_k s_k\|^2}{s_k^T B_k s_k} \right]^{1/q} \leq \frac{\Omega}{q^2} \sum_{k=k_0+1}^{k_0+q} \frac{\|B_k s_k\|^2}{s_k^T B_k s_k} \\
&\leq \frac{\Omega}{q^2} \sum_{k=0}^{k_0+q} \frac{\|B_k s_k\|^2}{s_k^T B_k s_k} \leq \frac{\Omega}{q^2} \left( \frac{\Delta}{\delta} \text{tr}(B_0) + \frac{\Delta}{\delta} (k_0 + q) + \frac{1}{\delta} \right),
\end{aligned} \tag{5.14}$$

where the last inequality follows from Proposition 5.2. Now, considering  $q \rightarrow \infty$ , we get a contradiction because of Proposition 5.3 which shows that the left-hand side of the above inequality (5.14) is greater than a positive constant. Therefore, (5.11) is true.  $\blacksquare$

Observe that the global convergence of the algorithm DSBFGS with  $\gamma_k$  given by (3.10) where  $\beta_k = |s_k^T g_{k+1}|$  bounded from below and with  $\delta_k$  given by (4.2) lower and upper bounded is proved in very general conditions without the convexity assumption of function  $f$ . This is the best result we can obtain under general assumptions that the function  $f$  is bounded from below and the line search is based on the inexact Wolfe line search conditions (1.4) and (1.5) and without the convexity assumption on  $f$ . Moreover, the above results can be obtained for any positive value for the parameter  $\beta_k$  tending to zero. The superlinear convergence of the scaled BFGS method (2.1) with the scaling parameters  $\gamma_k$  and  $\delta_k$  given by (3.10) and (4.2) respectively can be proved by using a tool and the results presented by Byrd and Nocedal [15] and Dennis and Moré [18, 19] (see also [29]). If the Hessian matrix  $\nabla^2 f(x)$  of the minimizing function  $f$  is Lipschitz continuous at the optimal solution  $x^*$  of the problem (1.1), then for any positive definite matrix  $B_0$  the modified BFGS method (2.1) with the scaling parameters given by (3.10) and (4.2), and the line search satisfying the inexact Wolfe line search conditions (1.4) and (1.5), generates a sequence  $\{x_k\}$  which converges to  $x^*$  superlinearly. This result is obtained under very general assumptions that  $f$  is twice continuously differentiable near  $x^*$ ,  $\{x_k\}$  converges to  $x^*$  where  $\nabla f(x^*) = 0$ ,  $\nabla^2 f(x^*)$  is positive definite and  $\nabla^2 f(x)$  is Lipschitz continuous, again without convexity assumption on  $f$  (see also [29]).

**Remark 5.3.** The scaling factor  $\delta_k$  in (4.2) is determined only from the equation  $\text{tr}(B_{k+1}) = n$ , i.e. using only the trace operator. This is not a limitation: if the stepsizes  $\alpha_k$  tend to zero, then, as proved by Byrd, Nocedal and Yuan [14] this is due to the existence of very small eigenvalues in  $B_k$ , which cannot be monitored by the trace operator. However, the BFGS update formula has a strong self-correcting property with respect to the determinant which can be used to show that, in fact,  $\alpha_k$  is bounded away from zero in mean. From (5.8) we see that when  $s_i^T B_i s_i$  is small relative to  $y_i^T s_i$ , then the determinant increases, showing that the small curvature of the model of the minimizing function is corrected, thus increasing some eigenvalues satisfying the condition  $\text{tr}(B_{k+1}) = n$ .

## 6. Numerical results and comparisons

In this section we present some numerical results with a Fortran implementation of the scaled BFGS algorithms we presented above. For this, the algorithm DSBFGS is particularized as follows: BFGS (DSBFGS with  $\delta_k = 1$  and  $\gamma_k = 1$ , i.e. the standard BFGS), BFGSC (DSBFGS with  $\delta_k = 1$  and  $\gamma_k$  given by (3.9), i.e. the scaled BFGS given by Cheng and Li [16]), BFGSB (DSBFGS with  $\delta_k = 1$  and  $\gamma_k$  given by (3.6), i.e. the scaled BFGS proposed by Biggs [9, 10]), BFGSY (DSBFGS with  $\delta_k = 1$  and  $\gamma_k$  given by (3.8), i.e. the scaled BFGS suggested by Yuan [50]), BFGSA (DSBFGS with  $\delta_k = 1$  and  $\gamma_k$  given by (3.10) with  $\beta_k = |s_k^T g_{k+1}|$ , i.e. the scaled BFGS proposed by Andrei [6]), BFGSD (DSBFGS with  $\delta_k$  and  $\gamma_k$  are given by (4.2) and (3.10), with  $\beta_k = |s_k^T g_{k+1}|$ , respectively, i.e. the scaled BFGS given by Andrei [7]), NOYA (DSBFGS with  $\delta_k = y_k^T s_k / s_k^T B_k s_k$  and  $\gamma_k = 1$ , i.e. the scaled BFGS given by Nocedal and Yuan [36]) and LIAO (scaled BFGS by Liao [30], given (3.11) and (3.12)).

All the algorithms implement the Wolfe line search conditions with  $\sigma = 0.8$  and  $\rho = 0.0001$ . The iterations are stopped if the inequality  $\|g_k\|_\infty \leq 10^{-5}$  is satisfied, where  $\|\cdot\|_\infty$  is the maximum absolute component of a vector or if the number of iterations exceeds 1000. In all the algorithms, for all the problems, the initial matrix  $H_0 = I$ , i.e. the identity matrix. For each method, except the method of Liao given by (3.11) and (3.12), in order to get the search direction we do not solve the system  $B_k d = -g_k$  to get  $d_k$ . Instead, we use the inverse updating formula (2.2). For the scaled BFGS methods by Biggs [9, 10] and Yuan [50],  $\gamma_k$  given by (3.6) and (3.8) respectively is restricted in the interval  $[0.01, 100]$ . Besides, at the very first iteration of these methods the scaling is not applied. All the codes were written in double precision Fortran and compiled with f77 (default compiler settings) on an Intel Pentium 4, 1.8GHz workstation. All the codes are authored by Andrei.

For a start, we present a simple example which illustrates the main elements of running the BFGSD algorithm. Consider the problem:

$$\min f(x) = \sum_{i=1}^n (\exp(x_i) - \sqrt{i} x_i),$$

where  $n=10$  and  $x_0=[1,1,\dots,1]$ . For this problem  $f(x_0)=4.71454$  and the BFGSD algorithm gives a local optimal solution for which  $f(x^*)=3.19505$  in 8 iterations and 42 evaluations of the function  $f$  and of its gradient.

Table 1 presents: the eigenvalues  $\lambda_1, \dots, \lambda_{10}$  of the Hessian approximation given by (2.1); the scaling factors  $\gamma_k$  and  $\delta_k$  given by (3.10) and (4.2) with  $\beta_k = |s_k^T g_{k+1}|$ , respectively; as well as the evolution of the elements  $\|B_k s_k\|^2$  and  $s_k^T B_k s_k$  for  $k=1, \dots, 8$ .

**Table 1.** Characteristics of the BFGSD algorithm.  
Eigenvalues of Hessian approximation (2.1),  $\delta_k$  given by (4.2),  $\gamma_k$  given by (3.10),

$\ B_k s_k\ ^2$ and $s_k^T B_k s_k$ .								
$k$	1	2	3	4	5	6	7	8
$\lambda_1$	0.8532	0.6423	0.6414	0.5303	0.4471	0.8606	1.5591	1.5288
$\lambda_2$	1.0094	1.0227	1.0075	0.9921	1.5161	0.9380	0.6479	1.3073
$\lambda_3$	1.0094	1.0227	1.0075	0.9921	1.0637	0.9380	1.1881	0.6181
$\lambda_4$	1.0094	1.0227	1.0075	0.9921	1.0061	0.9380	0.8883	0.8711
$\lambda_5$	1.0094	1.0227	1.0075	0.9921	0.9987	0.9388	0.9893	0.9821
$\lambda_6$	1.0094	1.0227	1.0075	0.9921	0.9951	0.9395	0.9496	0.9427
$\lambda_7$	1.0094	1.0227	1.0075	0.9966	0.9933	0.9463	0.9451	0.9380
$\lambda_8$	1.0094	1.0227	1.0185	1.0030	0.9933	0.9656	0.9443	0.9375
$\lambda_9$	1.0094	1.0749	1.0729	1.0667	0.9933	1.0090	0.9442	0.9372
$\lambda_{10}$	1.0713	1.1239	1.2223	1.4431	0.9933	1.5262	0.9442	0.9372
$\delta_k$	1.0094	1.0131	0.9851	0.9847	1.0012	0.9443	1.0065	0.9926
$\gamma_k$	0.4193	0.4880	0.5943	0.5338	0.4343	0.9285	0.4195	0.4488
$\ B_k s_k\ ^2$	1	0.356e-1	0.5203-2	0.553e-3	0.321e-4	0.249e-4	0.236e-6	0.207e-7
$s_k^T B_k s_k$	1	0.417e-1	0.809e-2	0.845e-3	0.554e-4	0.531e-4	0.224e-6	0.223e-7

An attractive feature of the BFGSD algorithm, where  $\gamma_k$  and  $\delta_k$  are given by (3.10) and (4.2) with  $\beta_k = |s_k^T g_{k+1}|$ , respectively, which we see in Table 1, is that along the iterations, the eigenvalues of the Hessian approximation (2.1) are all positive and clustered. In fact, the Hessian approximation (2.1) has a special eigenvalue structure that occurs in BFGSD: there are some large eigenvalues that may or may not be located near each other, as well as some smaller eigenvalues located more or less near 1, all satisfying the condition  $\sum_{i=1}^n \lambda_i = n$ . Observe that this structure of the eigenvalues of the Hessian approximation (2.1) is very similar to the structure of the eigenvalues encountered in conjugate gradient algorithms where the approximation to the inverse Hessian is restarted as identity matrix at every step (see [45]).

From Table 1 we see that  $\delta_k$  computed as in (4.2) is close to 1, as proved in Proposition 4.1.

Observe that along the iterations,  $\|B_k s_k\|^2$  and  $s_k^T B_k s_k$  are of the same order of magnitude, both of them tending to zero.

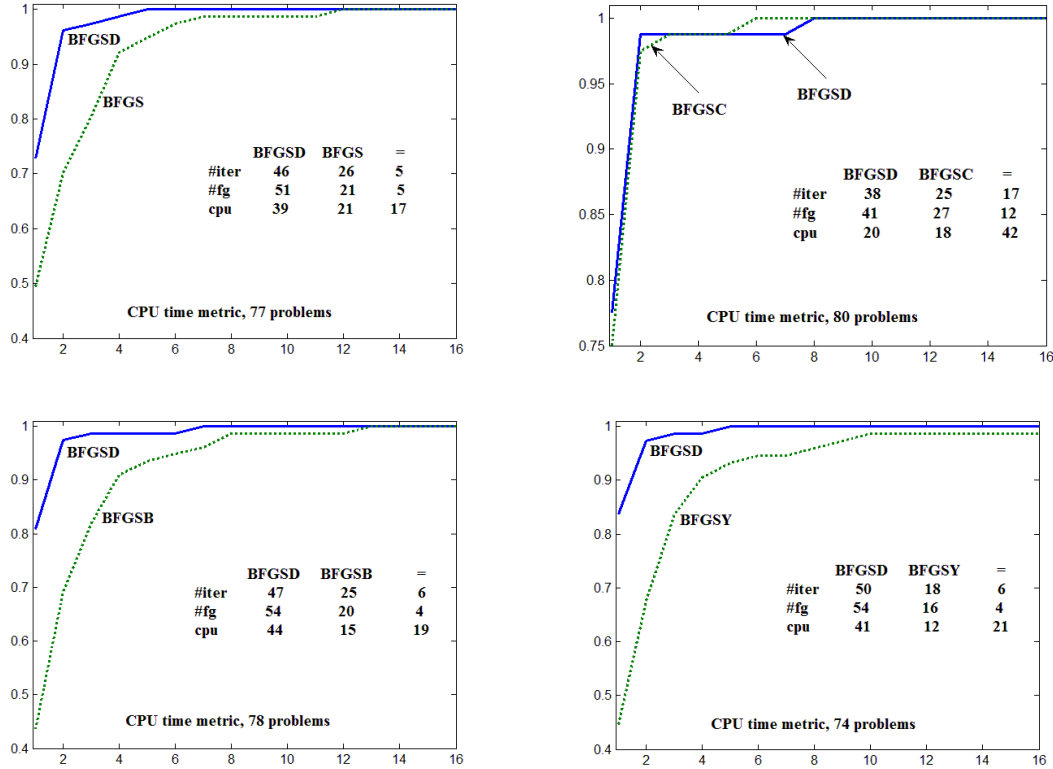
In the following, we considered a number of 80 unconstrained optimization test problems of medium size ( $n=100$  variables), described in [4]. The algorithms which we compare in these numerical experiments find local solutions. Therefore, the comparisons of the algorithms are given in the following context. Let  $f_i^{ALG1}$  and  $f_i^{ALG2}$  be the optimal value found by ALG1 and ALG2 for problem  $i=1,\dots,80$ , respectively. We say that, in the particular problem  $i$ , the performance of ALG1 was better than the performance of ALG2 if:

$$|f_i^{ALG1} - f_i^{ALG2}| < 10^{-3} \quad (6.1)$$

and the number of iterations (#iter), or the number of function-gradient evaluations (#fg), or the CPU time of ALG1 was less than the number of iterations, or the number of function-gradient evaluations, or the CPU time corresponding to ALG2, respectively.

In the first set of numerical experiments we compare BFGSD versus BFGS, BFGSC, BFGSB and BFGSY. For BFGSC, BFGSB and BFGSY the search direction is computed as in (1.7) where  $H_{k+1}$  is updated as in (2.2) with  $\delta_k=1$  and the corresponding values of  $\gamma_k$ . For the standard BFGS algorithm the search direction is determined as in (1.7) where the approximation to the inverse Hessian is updated as in (1.8).

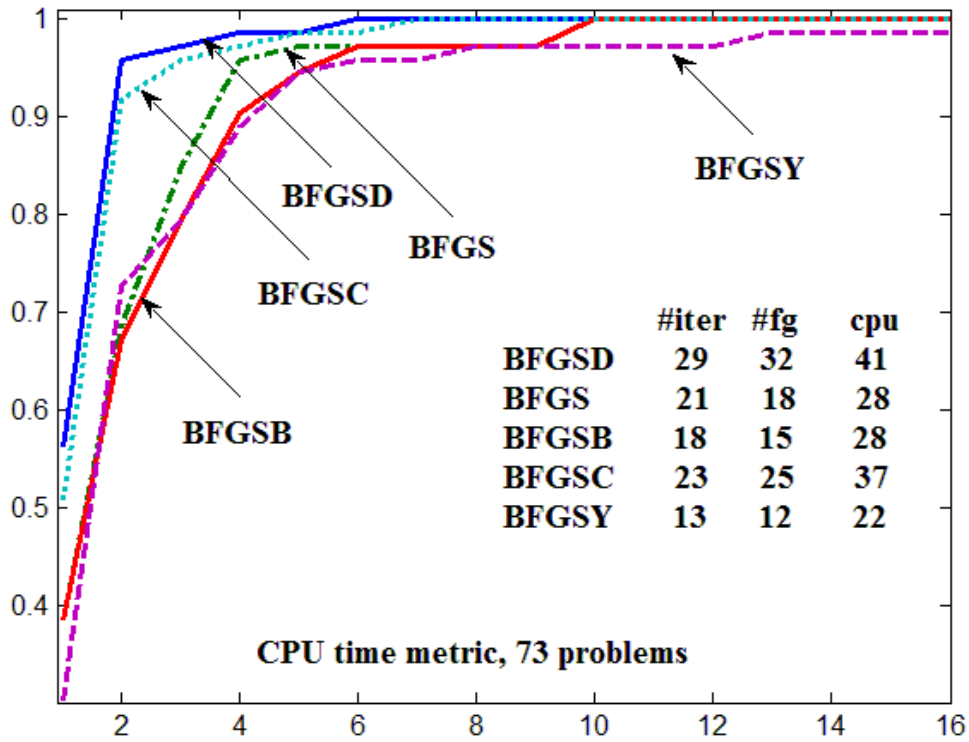
Figure 1 presents the Dolan and Moré [21] performance profiles of these algorithms for this set of unconstrained optimization test problems based on the CPU time metric. For example, when comparing BFGSD versus BFGS (standard BFGS algorithm), subject to the number of iterations, we see that BFGSD was better in 46 problems (i.e. it achieved the minimum number of iterations in 46 problems), BFGS was better in 26 problems. Both of them achieved the same number of iterations in 5 problems, etc. Out of 80 problems considered in this set of numerical experiments only for 77 does the criterion (6.1) hold.



**Fig.1.** Performance profiles of BFGSD versus BFGS, BFGSC, BFGSB and BFGSY.

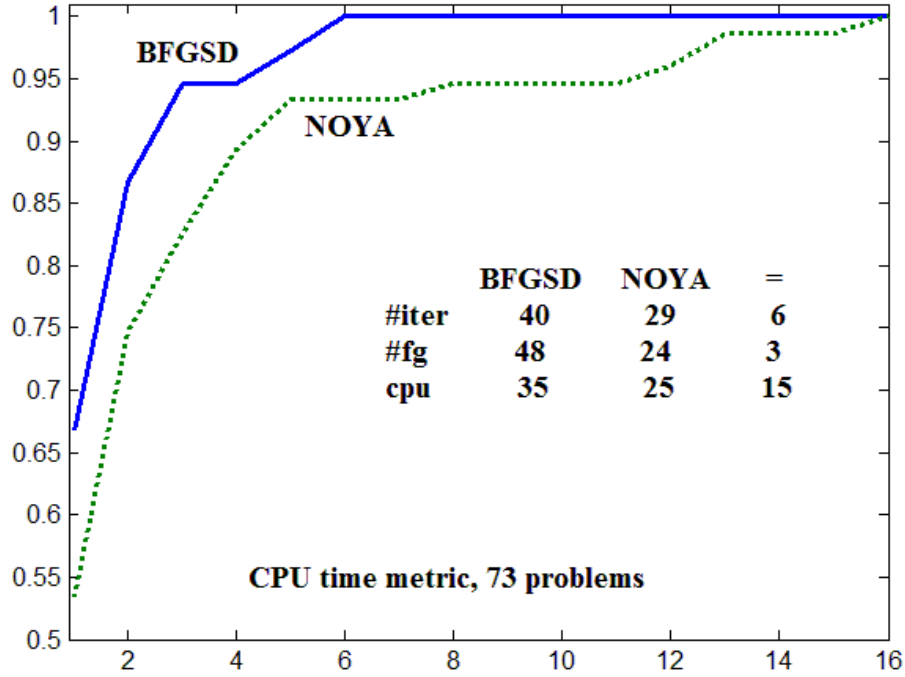
From the performance profiles given in Figure 1 we see that BFGSD is top performer against BFGS, BFGSB, BFGSC and BFGSY algorithms. Since all these codes use the same Wolfe line search and the same stopping criterion, they differ only in their choice of the search direction. The percentage of the test problems for which a method is the fastest is given on the left axis of the plot. The right side of the plot gives the percentage of the test problems that were successfully solved by these algorithms. Mainly, the left side is a measure of the efficiency of an algorithm; the right side is a measure of the robustness. From Figure 1 we see that BFGSD is top performer versus the classical BFGS and the scaled BFGS algorithms (BFGSB, BFGSC, BFGSY) considered in this numerical study and the differences are significant.

Figure 2 presents the performance profiles of all these 5 modified BFGS methods subject to the CPU computing time metric. From Figure 2 we see that subject to the CPU time metric the BFGSD algorithm is top performer versus the standard BFGS algorithm and versus the scaled BFGSB, BFGSC and BFGSY algorithms. Observe that BFGSD and BFGSC are grouped, having better performances versus the other ones.



**Fig. 2.** Performance profile of BFGSD, BFGS, BFGSB, BFGSC and BFGSY.

In the second set of numerical experiments we compare the scaled BFGSD algorithm versus NOYA - the self-scaled BFGS algorithm by Nocedal and Yuan [36], where the approximation of the Hessian  $B_{k+1}$  is computed as in (2.1) with  $\delta_k = y_k^T s_k / s_k^T B_k s_k$  and  $\gamma_k = 1$ . Figure 3 presents the performance profile of BFGSD versus NOYA subject to CPU time metric. From Figure 3 we see that the BFGSD algorithm is top performer versus NOYA.

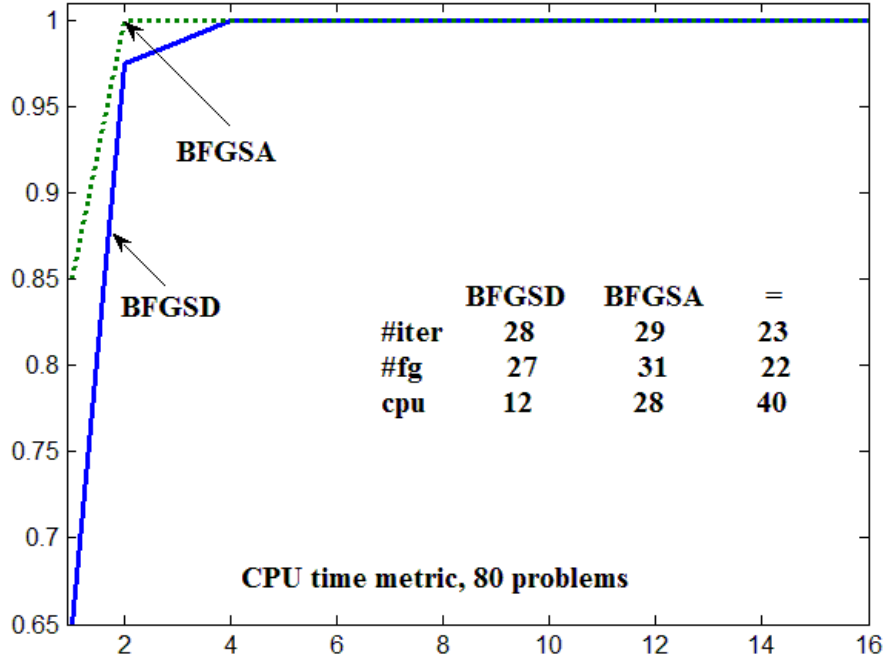


**Fig. 3.** Performance profile of BFGSD versus NOYA.

In their study, Nocedal and Yuan proved that the self-scaled BFGS algorithm NOYA, with inexact line search, is globally convergent on general convex functions. However, the main drawback of this algorithm is that for achieving superlinear convergence it might need to evaluate the minimizing function twice per iteration, even very near the solution [36]. The scaling of the first two terms of  $B_{k+1}$  matrix with  $\delta_k = y_k^T s_k / s_k^T B_k s_k$ , like in NOYA algorithm, leads us to disappointing numerical results. This is consistent with the analysis given by Nocedal and Yuan [36] and Shanno and Phua [46]. On the other hand, in our study on the scaled BFGS algorithm BFGSD we emphasize that both parameters  $\gamma_k$  and  $\delta_k$  are important in the economy of the algorithm:  $\delta_k$  is computed to cluster the eigenvalues of  $B_{k+1}$  and  $\gamma_k$  is responsible for shifting the large eigenvalues to the left. These are the main motivations that DSBFGS has better performances versus NOYA.

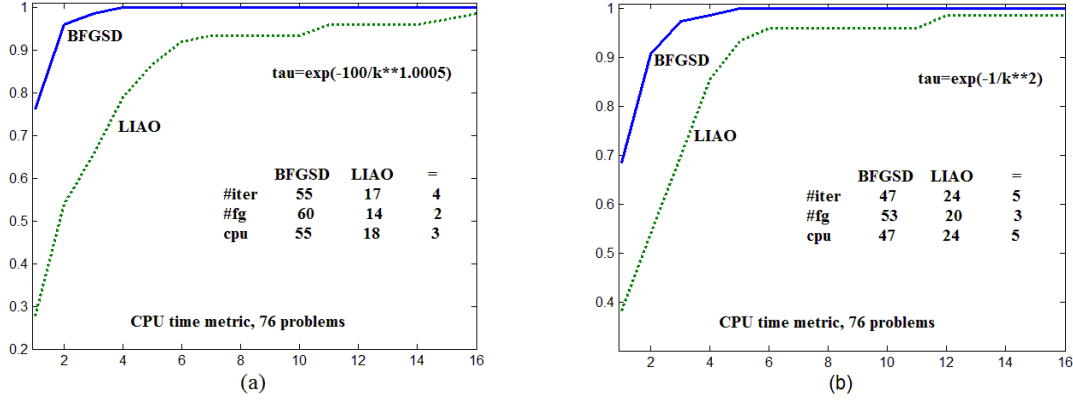
In the third set of numerical experiments we compare the scaled BFGSD algorithm versus BFGSA. In BFGSA the search direction is determined like in (1.7) where the inverse approximation to the Hessian is computed as in (2.2), with  $\delta_k = 1$  and  $\gamma_k$  given by (3.10) with  $\beta_k = |s_k^T g_{k+1}|$  [6]. Figure 4 shows the performance profiles of these algorithms subject to CPU computing time. Observe that BFGSA is top performer versus BFGSD, being much more efficient. In Proposition 4.1 we proved that  $\delta_k$  is close to 1. Therefore, in the economy of the BFGSD algorithm  $\delta_k$ , which scale the first two terms of the BFGS update, is selected as in (4.2) to cluster the eigenvalues of the scaled BFGS matrix in such a way that their sum to be equal to the dimension of the problem. On the other hand, in BFGSA the spectrum of the scaled BFGS matrix is free. Oren and Luenberger [38] showed that in order to guarantee that the BFGS update  $B_{k+1}$  will have a lower condition number than  $B_k$ , the interval spanned by the eigenvalues of  $B_k$

must contain the unity. But in our numerical experiments we noticed that along the iterations the spectrum of the BFGS update matrix generated by BFGSA always contain unity. On the other hand, in BFGSA the scaling factor  $\gamma_k$  is selected as in (3.10) with  $\beta_k = |s_k^T g_{k+1}|$  in order to be a diagonal preconditioner of  $\nabla^2 f(x_{k+1})$  and also to minimize the conjugacy condition  $d_{k+1}^T y_k = -s_k^T g_{k+1}$ . These are the major arguments that BFGSA is superior versus BFGSD.



**Fig. 4.** Performance profile of BFGSD versus BFGSA

In the last set of numerical experiments we compare the scaled BFGSD algorithm versus LIAO - the scaled BFGS algorithm by Liao [30]. In the Liao algorithm the Hessian approximation  $B_{k+1}$  is computed as in (3.11), where the parameters  $\delta_k$  and  $\gamma_k$  are computed as in (3.12). Figure 5a presents the Dolan and Moré performance profiles of BFGSD versus LIAO with  $\tau_k = \exp(-100/k^{1.0005})$ . Figure 5b presents the performance profiles of BFGSD versus LIAO with  $\tau_k = \exp(-1/k^2)$ . We observed that if  $\tau_k$  is small, like in the LIAO algorithm with  $\tau_k = \exp(-100/k^{1.0005})$ , then the algorithm takes  $\delta_k = s_k^T B_k s_k / (s_k^T B_k s_k + y_k^T s_k)$  and  $\gamma_k = y_k^T s_k / (s_k^T B_k s_k + y_k^T s_k)$ , as it is specified in (3.12). If  $\tau_k$  is large, like in the LIAO algorithm with  $\tau_k = \exp(-1/k^2)$ , then the algorithm selects  $\delta_k = \tau_k$  and  $\gamma_k = 1$ , as it is recommended by (3.12). From Figure 5 we see that BFGSD algorithm is top performer versus both variants of LIAO. The performances of the scaled BFGS algorithm LIAO are very sensitive to the values of the parameter  $\tau_k$ . Both these values of  $\tau_k$  has been suggested by Liao [30], but some other values could be tried leading us to similar results. It seems that BFGS update has a remarkable robustness subject to the scaling the last two terms in (1.6).



**Fig. 5.** Performance profile of BFGSD versus LIAO.

From (3.11) we get:

$$tr(B_{k+1}) = tr(B_k) - \delta_k \frac{\|B_k s_k\|^2}{s_k^T B_k s_k} + \gamma_k \frac{\|y_k\|^2}{y_k^T s_k}. \quad (6.2)$$

From (3.12) we see that if  $s_k^T B_k s_k > y_k^T s_k$ , then  $0 < \gamma_k < \delta_k < 1$ . Therefore, the second term in (6.2) which shifts the eigenvalues to the left is almost the same as the second term in (1.9), but the third term in (6.2) which shifts the eigenvalues to the right is much smaller than the third term in (1.9). In this case, the LIAO algorithm better corrects the large eigenvalues than the standard BFGS does. In comparison, in BFGSD, the large eigenvalues are not only shifted to the left by means of  $\gamma_k$  selected as in (3.10), but they are clustered by a proper selection of  $\delta_k$  as in (4.2). This is the reason why BFGSD is more efficient and more robust than LIAO (see Fig. 5).

Since in LIAO the search direction  $d_{k+1}$  is computed as solution of the system  $B_{k+1} d_{k+1} = -g_{k+1}$ , we generated a Fortran version of BFGSD code where the search direction is computed as solution of the system  $B_{k+1} d_{k+1} = -g_{k+1}$  to compare it versus the LIAO algorithm. Therefore, unlike the previous numerical experiments, in this comparison, both in BFGSD and in LIAO the search direction  $d_{k+1}$  is computed as solution of the system  $B_{k+1} d_{k+1} = -g_{k+1}$ . From Figure 5 we see that BFGSD is top performer versus LIAO and the difference is significant subject to the efficiency and robustness of the algorithms. Since these codes use the same Wolfe line search and the same stopping criterion, they differ only in their choice of the search direction. Again, observe that the numerical results with LIAO are disappointing. This is because in LIAO the modified (scaled) BFGS update is obtained by a simple symmetrization procedure from a rank one update (see [30]).

As a byproduct, it is worth saying that the BFGSD algorithm where the search direction is computed as  $d_{k+1} = -H_{k+1} g_{k+1}$  is much faster than its version where  $d_{k+1}$  is computed as solution of the system  $B_{k+1} d_{k+1} = -g_{k+1}$ .

## 5. Conclusions

Plenty of scaled BFGS algorithms are known. In this paper we suggested a new double parameter scaled BFGS method where the first two terms in standard BFGS update are scaled with a positive parameter and the third term is scaled with another positive one. In our algorithm the factor scaling the first two terms of the standard BFGS update is selected to cluster the



eigenvalues of the scaled BFGS update. On the other hand, the factor scaling the third term is determined to shift the large eigenvalues to the left. For general functions we proved that the algorithm with inexact line search is globally convergent under the very reasonable condition that the scaling parameters are bounded. Numerical results using a limited number of 80 unconstrained optimization test problems of different structures and complexities show that this scaled BFGS update with two parameters is more efficient versus the standard BFGS algorithm as well as versus some other well known scaled BFGS algorithms, including those by Biggs [9, 10], Cheng and Li [16], Liao [30], Nocedal and Yuan [36] and Yuan [50]. The conclusion of this study is that scaling the first two terms of the standard BFGS update has an important effect on the performances of the scaled BFGS algorithm. The most important is the scaling of the third term of the standard BFGS update (see [6]). The scaling of the third term with  $\gamma_k < 1$  will push down to the left the eigenvalues of the scaled BFGS update, thus obtaining a better structure of the eigenvalues than the one of the standard BFGS or of some other scaled BFGS methods. This is in agreement with the computational performances of the BFGSA algorithm in which only the third term of the right hand side term of the BFGS update is scaled. In fact, BFGSA proved to be the most efficient among the scaled BFGS algorithms discussed in this paper.

The main lesson we get from this study is that scaling the terms of the standard BFGS update may lead to algorithms that are more efficient than the standard BFGS algorithm. However, selecting the values for the scaling factors is not easy to guess. In our algorithm, for scaling factors determination we implemented the idea of clustering the eigenvalues of the iteration matrix and of shifting its large eigenvalues to the left by using the trace operator. Some other principles may be used in which the scaling factors are determined by using the determinant of the iteration matrix, or a combination of these two operators (trace and determinant). Another idea is to scale the terms of the standard BFGS update at some selected iterations, for example only during the first few iterations. Anyway, the BFGS quasi-Newton methods continue to be full of surprises, always having more room for improving their numerical performances.

## References

- [1] Al-Baali, M.: Analysis of a family of self-scaling quasi-Newton methods. Technical Report, Department of Mathematics and Computer Science, United Arab Emirates University, (1993)
- [2] Al-Baali, M.: Global and superlinear convergence of a class of self-scaling methods with inexact line searches. *Computational Optimization and Applications* 9, 191-203 (1998)
- [3] Al-Baali, M.: Numerical experience with a class of self-scaling quasi-Newton algorithms. *Journal of Optimization Theory and Applications* 96, 533-553 (1998)
- [4] Andrei, N.: An unconstrained optimization test functions collection. *Advanced Modeling and Optimization – An Electronic International Journal* 10, 147-161 (2008)
- [5] Andrei, N.: Eigenvalues versus singular values study in conjugate gradient algorithms for large-scale unconstrained optimization. *Optimization Methods and Software*, 32, 534-551 (2017)
- [6] Andrei, N.: An adaptive scaled BFGS method for unconstrained optimization. *Numerical Algorithms*. DOI: 10.1007/s11075-017-0321-1.
- [7] Andrei, N.: A double parameter scaled BFGS method for unconstrained optimization. ICI Technical Report, Bucharest, May 2, 2017.
- [8] Barzilai, J., Borwein, J.M.: Two-points step size gradient methods. *IMA Journal of Numerical Analysis* 8, 141-148 (1988)
- [9] Biggs, M.C.: Minimization algorithms making use of non-quadratic properties of the objective function. *Journal of the Institute of Mathematics and Its Applications* 8, 315-327 (1971)

- [10] Biggs, M.C.: A note on minimization algorithms making use of non-quadratic properties of the objective function. *Journal of the Institute of Mathematics and Its Applications* 12, 337-338 (1973)
- [11] Bongartz, I., Conn, A.R., Gould, N.I.M., Toint, Ph.L.: CUTER: constrained and unconstrained testing environments. *ACM Trans. Math. Software* 21, 123-160 (1995)
- [12] Broyden, C. G.: The convergence of a class of double-rank minimization algorithms. I. General considerations. *J. Inst. Math. Appl.* 6, 76-90 (1970)
- [13] Byrd, R.H., Liu, D.C., Nocedal, J.: On the behavior of Broyden's class of quasi-Newton methods. *SIAM J. Optim.*, 2, 533-557 (1992)
- [14] Byrd, R., Nocedal, J., Yuan, Y.: Global convergence of a class of quasi-Newton methods on convex problems. *SIAM Journal on Numerical Analysis* 24, 1171-1189 (1987)
- [15] Byrd, R., Nocedal, J.: A tool for the analysis of quasi-Newton methods with application to unconstrained minimization. *SIAM Journal on Numerical Analysis* 26, 727-739 (1989)
- [16] Cheng, W.Y., Li, D.H.: Spectral scaling BFGS method. *Journal of Optimization Theory and Applications* 146, 305-319 (2010)
- [17] Dai, Yu-Hong.: Convergence properties of the BFGS Algorithm. *SIAM J. Optim.* 13 (3), 693-701 (2002)
- [18] Dennis, J.E., Moré, J.J.: A characterization of superlinear convergence and its application to quasi-Newton methods. *Mathematics of Computation* 28, 549-560 (1974)
- [19] Dennis, J.E., Moré, J.J.: Quasi-Newton methods, motivation and theory. *SIAM Review* 19, 46-89 (1977)
- [20] Dixon, L.C.W.: Variable metric algorithms: necessary and sufficient conditions for identical behavior on nonquadratic functions. *Journal of Optimization Theory and Applications* 10, 34-40 (1972)
- [21] Dolan, E.D., Moré, J.J.: Benchmarking optimization software with performance profiles. *Mathematical Programming* 91, 201-213 (2002)
- [22] Fletcher, R.: A new approach to variable metric algorithms. *The Computer Journal* 13, 317-322 (1970)
- [23] Fletcher, R.: An overview of unconstrained optimization. In: *Algorithms for Continuous Optimization: The State of the Art*, E. Spedicato (Ed.), Kluwer Academic Publishers, Boston, 109-143 (1994)
- [24] Gill, P.E., Leonard, M.W.: Reduced-Hessian quasi Newton methods for unconstrained optimization. *SIAM Journal on Optimization* 12, 209-237 (2001)
- [25] Goldfarb, D.: A family of variable metric methods derived by variation mean. *Mathematics of Computation* 23, 23-26 (1970)
- [26] Griewank, A.: The global convergence of partitioned BFGS on problems with convex decompositions and Lipschitzian gradients. *Mathematical Programming* 50, 141-175 (1991)
- [27] Hestenes, M.R., Stiefel, E.L., *Methods of conjugate gradients for solving linear systems*, J. Research Nat. Bur. Standards, 49 (1952), pp.409-436.
- [28] Lalee, M., Nocedal, J.: Automatic column scaling strategies for quasi-Newton methods. Report No. NAM 04, Department of Electrical Engineering and Computer Science, Northwestern University, Evanston, IL (1991)
- [29] Li, D-H, Fukushima, M.: A modified BFGS method and its global convergence in nonconvex minimization. *Journal of Computational and Applied Mathematics* 129, 15-35 (2001)
- [30] Liao, A.: Modifying BFGS method. *Operations Research Letters* 20, 171-177 (1997)
- [31] Loewner, C.: Advanced matrix theory. Lecture Notes, Stanford University, Winter (1957)
- [32] Luenberger, D.G.: *Introduction to Linear and Nonlinear Programming*. Addison-Wesley, Reading (1973)
- [33] Luenberger, D.G.: *Linear and Nonlinear Programming*. Addison-Wesley, Reading, MA, 1984 (2<sup>nd</sup> ed.)

- [34] Mascarenhas, W.F.: The BFGS method with exact line searches fails for non-convex objective functions. *Mathematical Programming, Ser. A*, 99, 49-61 (2004)
- [35] Nocedal, J.: Theory of algorithms for unconstrained optimization. *Acta Numerica* 1, 199-242 (1992)
- [36] Nocedal, J., Yuan, Y.: Analysis of self-scaling quasi-Newton method. *Mathematical Programming* 61, 19-37 (1993)
- [37] Oren, S.S.: Perspectives on self-scaling variable metric algorithms. *Journal of Optimization Theory and Applications*. 37, 137-147 (1982).
- [38] Oren, S.S. Luenberger, G.G.: Self-scaling variable metric (SSVM) algorithms. Part I: criteria and sufficient conditions for scaling a class of algorithms. *Management Science* 20, 845-862 (1974)
- [39] Pearson, J.D.: Variable metric methods for minimization. *Computer Journal*, 12, 171-178 (1969).
- [40] Powell, M.J.D.: On the convergence of the variable metric algorithm. *Journal of the Institute of Mathematics and its Applications* 7, 21-36 (1971)
- [41] Powell, M.J.D.: Some global convergence properties of a variable metric algorithm for minimization without exact line search. In: R.W. Cottle and C.E. Lemke (Eds.) *Nonlinear Programming, SIAM-AMS Proceedings*, vol. IX, pp. 53-72, SIAM Philadelphia, PA (1976)
- [42] Powell, M.J.D.: How bad are the BFGS and DFP methods when the objective function is quadratic? *Math. Programming*, 34, 34-47 (1986)
- [43] Powell, M.J.D.: Updating conjugate directions by the BFGS formula. *Mathematical Programming* 38, 693-726 (1987)
- [44] Shanno, D.F.: Conditioning of quasi-Newton methods for function minimization. *Mathematics of Computation* 24, 647-656 (1970)
- [45] Shanno, D.F.: Conjugate gradient methods with inexact searches. *Mathematics of Operations Research*, 3, 244-256 (1978)
- [46] Shanno, D.F., Phua, K.H.: Matrix conditioning and nonlinear optimization. *Mathematical Programming* 14, 149-160 (1978)
- [47] Siegel, D.: Modifying the BFGS update by a new column scaling technique. Technical Report, Department of Applied Mathematics and Theoretical Physics, Cambridge University, Cambridge (1991)
- [48] Sun, W., Yuan, Y.X.: *Optimization theory and methods. Nonlinear programming*. Springer Science + Business Media, New York (2006)
- [49] Wang, H.J. Yuan, Y.: A quadratic convergence method for one-dimensional optimization. *Chinese Journal of Operations Research* 11, 1-10 (1992)
- [50] Yuan, Y.: A modified BFGS algorithm for unconstrained optimization. *IMA Journal Numerical Analysis* 11, 325-332 (1991)
- [51] Zhang, J.Z., Xu, C.X.: Properties and numerical performance of quasi-Newton methods with modified quasi-Newton equations. *J. Comput. Appl. Math.* 139, 269-278 (2001)
- [52] Wolfe, P.: Convergence conditions for ascent methods. *SIAM Review*, 11, 226-235 (1969)
- [53] Wolfe, P.: Convergence conditions for ascent methods. II: Some corrections. *SIAM Review*, 13, 185-188 (1971)