A double parameter scaled BFGS method for unconstrained optimization Preliminary computational results.

Neculai Andrei[1]
Research Institute for Informatics, Center for Advanced Modeling and Optimization,
8-10 Averescu Avenue, Bucharest 1, Romania

September 11, 2017

Abstract. A double parameter scaled BFGS method for unconstrained optimization is presented. In this method, the first two terms of the known BFGS update formula are scaled with a positive parameter while the third one is scaled with another positive parameter. These parameters are selected in such a way as to improve the eigenvalues structure of the BFGS update. 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. Under the inexact Wolfe line search, the global convergence of the double parameter scaled BFGS method 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 double parameter scaled BFGS method is more efficient than the standard BFGS update or than some other scaled BFGS methods.

Keywords: Unconstrained optimization . Scaled BFGS method . Self-correcting quality . Global convergence . Numerical comparisons.

Mathematics Subject Classification (2010) 49M7. 49M10. 65K05. 90C30

1. Introduction

Let $f: R^n \to R$ be a continuously differentiable function bounded from below and consider the following unconstrained minimization problem:

$$\min f(x), \tag{1.1}$$

where $x \in R^n$. Given an initial point $x_0 \in R^n$ and an initial approximation $B_0 \in R^{n \times n}$ to the Hessian of function f, symmetric and positive definite, for solving (1.1) the well known quasi-Newton BFGS method introduced by Broyden [1], Fletcher [2], Goldfarb [3] and Shanno [4], generates a sequence $\{x_k\}$ computed by the scheme:

$$x_{k+1} = x_k + \alpha_k d_k, \tag{1.2}$$

$k = 0,1,\ldots,$ where d_k is the BFGS search direction obtained as solution of the linear algebraic system

$$B_k d_k = -g_k, \tag{1.3}$$

E-mail address: nandrei@ici.ro
[1] Dr. Neculai Andrei is full member of Academy of Romanian Scientists, Splaiul Independenţei nr. 54, Sector 5, Bucharest, Romania.

and g_k is the gradient $\nabla f(x_k)$ of f at x_k. In (1.3) the matrix B_k is the BFGS approximation to the Hessian $\nabla^2 f(x_k)$ of f at x_k, being updated by the classical formula:

$$B_{k+1} = B_k - \frac{B_k s_k s_k^T B_k}{s_k^T B_k s_k} + \frac{y_k y_k^T}{y_k^T s_k}, \tag{1.4}$$

$k = 0,1,\ldots,$ where $s_k = x_{k+1} - x_k$ and $y_k = g_{k+1} - g_k$. An important property of the BFGS updating formula (1.4), which we call *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 α_k in (1.2) is determined by the Wolfe line search conditions [5, 6]:

$$f(x_k + \alpha_k d_k) \le f(x_k) + \sigma \alpha_k g(x_k)^T d_k, \tag{1.5}$$

$$g(x_k + \alpha_k d_k)^T d_k \ge \rho g(x_k)^T d_k, \tag{1.6}$$

where the positive constants σ and ρ satisfy $0 < \sigma < \rho < 1$. We note that the condition $y_k^T s_k > 0$ is also guaranteed to hold if the stepsize α_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 (1.3) is a descent direction of f at x_k, no matter whether the Hessian is positive definite or not.

The BFGS method proved to be one of the most efficient quasi-Newton methods for solving small and medium-size unconstrained optimization problems. An excellent presentation of the theoretical aspects concerning the properties and the convergence of this method were given by Dennis and Moré [7, 8]. At the same time, a deep analysis of the BFGS method and its variants was presented by Nocedal [9]. The BFGS method is fast and robust and it is currently used in innumerable optimization software for solving unconstrained or constrained optimization problems. The main results concerning its convergence property are as follows. For twice continuously differentiable convex functions with compact level sets, Powel [10] 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 [11, 12, 13, 14, 15]. On the other hand, for nonconvex problems under the exact line search, Mascarenhas [16] 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 methods for unconstrained optimization [9, 18].

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 [9]). 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 explained by Nocedal [9] is that it better corrects small eigenvalues than large ones. Powell [19] proved that BFGS with inexact Wolfe line search is globally superlinear convergent for convex problems. On the other hand, Byrd and Nocedal [12] 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 [11] 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 [20]. The sources of inefficiency of the BFGS method may be caused by a poor initial approximation to the

Hessian or, more importantly, by the ill-conditioning of the Hessian approximations along the iterations, thus leading to a poorly defined search direction.

In order to improve the performances of the BFGS method, the *self-scaling BFGS methods* have been derived, firstly suggested and analyzed for the minimization of the quadratic functions. Oren and Luenberger [21] scaled the Hessian approximation B_k before updating it, i.e. they replaced B_k by $\tau_k B_k$, where τ_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. Nocedal and Yuan [22] further studied the self-scaling BFGS method when $\tau_k = y_k^T s_k / s_k^T B_k s_k$, where $s_k = x_{k+1} - x_k$ and $y_k = g_{k+1} - g_k$, (see also Nocedal [9]). An extension of this self-scaling BFGS method was considered by Al-Baali [23], who introduced a simple modification: $\tau_k = \min\{1, \tau_k\}$. The numerical experiments in [23] showed that the modified self-scaling BFGS method is competitive versus the unscaled BFGS method. In the same line of efforts, Al-Baali [24] 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 τ_k. The global convergence and the local superlinear convergence of this class of self-scaling methods with inexact line search were proved by Al-Baali [24]. The numerical experiments with this restricted class of self-scaling quasi-Newton methods were reported by Al-Baali [25] on a set of small test unconstrained optimization problems up to 20 variables.

Many other modified BFGS methods were suggested. Using different function interpolation conditions, Biggs [26, 27] and Yuan [28] obtained some modified BFGS methods and proved their global convergence. The idea of their method was to scale the third term of the BFGS updating formula. The modified BFGS method by Yuan uses both the gradient and the function values information in one step. Another self-scaling modified BFGS method was suggested by Aiping Liao [29]. 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. The global convergence of this two parameters scaled BFGS modified method is proved by using a tool introduced by Byrd and Nocedal [12]. Another scaled BFGS method was proposed by Nocedal and Yuan [22], where the first two terms of the BFGS updating formula are scaled by the same factor $y_k^T s_k / s_k^T B_k s_k$. 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 [30]. A recent spectral scaling BFGS method was proposed by Cheng and Li [31]. In their method, the standard BFGS update is modified by introducing a positive scale factor γ_k to the third term of the BFGS updating formula, which is exactly the Barzilai and Borwein [32] 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 [28], Al-Baali [25], Zhang and Xu [33] 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 [34]. 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 [26, 27], Cheng and Li [31] and Yuan [28].

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 [19] and further developed by Lalee and Nocedal [35] and Siegel [36]. 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 [35] 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 [36] suggested scaling up the last l columns of Z_k, where l is an integer parameter.

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 some procedures for selection of the scaling parameters in scaled BFGS update as found in literature. Section 3 is devoted to detail a two parameter scaled BFGS update and the corresponding TPSBFGS algorithm. 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. Some properties of this algorithm are proved. The global convergence analysis of the double parameter scaled BFGS algorithm is presented in Section 4. The analysis is based on the developments presented in [12, 34] and [37]. We find that the double parameter scaled BFGS algorithm 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 5 some numerical results of the suggested double parameter scaled BFGS algorithm are presented by using 80 unconstrained optimization medium size test problems. At the same time, comparisons versus the standard BFGS algorithm, as well as versus some other scaled BFGS algorithms by Biggs [26, 27], Cheng and Li [31], Yuan [28], Nocedal and Yuan [22], Andrei [34] and Liao [29] are given. We have the computational evidence that our double parameter scaled BFGS algorithm is much more efficient and more robust than all these scaled BFGS algorithms. However, the scaled BFGS update by Andrei [34] is more efficient.

2. Selection of the Scaling Parameters in the BFGS update

One of the first scaled BFGS update was

$$B_{k+1} = B_k - \frac{B_k s_k s_k^T B_k}{s_k^T B_k s_k} + \gamma_k \frac{y_k y_k^T}{y_k^T s_k}, \tag{2.1}$$

where $\gamma_k > 0$ is a parameter. For the scaling parameter γ_k in (2.1) some values have been proposed in literature, as follows.

1) *Scaled BFGS with different interpolation conditions* (Biggs [26, 27] and Yuan [28]). Observe that the quasi-Newton step $d_k = -H_k g_k$ is a stationary point of the following problem:

$$\min_{d \in R^n} \phi_k(d) = f(x_k) + g_k^T d + \frac{1}{2} d^T B_k d. \tag{2.2}$$

Since for small d, $\phi_k(d) \approx f(x_k + d)$, it follows that the problem (2.2) is an approximation to the problem (1.1) near the current point x_k. From (2.2) we have that

$$\phi_k(0) = f(x_k), \quad \nabla\phi_k(0) = g(x_k), \tag{2.3}$$

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

$$\nabla\phi_k(x_{k-1}-x_k)=g(x_{k-1}). \tag{2.4}$$

Therefore $\phi_k(x-x_k)$ is a quadratic interpolation of $f(x)$ at x_k satisfying the above conditions (2.3) and (2.4).

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

$$s_{k-1}^T\nabla^2 f(x_k)s_{k-1}=4s_{k-1}^T g_k+2s_{k-1}^T g_{k+1}-6(f(x_{k-1})-f(x_k)). \tag{2.5}$$

Biggs [26, 27] considers the update (2.1) where the value of γ_k is chosen in such a way that the new approximate Hessian satisfies the reasonable condition

$$s_{k-1}^T B_k s_{k-1}=4s_{k-1}^T g_k+2s_{k-1}^T g_{k+1}-6(f(x_{k-1})-f(x_k)). \tag{2.6}$$

Therefore, the value of γ_k proposed by Biggs is

$$\gamma_k=\frac{6}{y_k^T s_k}(f(x_k)-f(x_{k+1})+s_k^T g_{k+1})-2. \tag{2.7}$$

For one-dimensional problems, Wang and Yuan [38] showed that the scaled BFGS (2.1) with γ_k given by (2.7) and without line search is R-linear convergent.

In the same line of research, Yuan [28] considered that the approximate function $\phi_k(d)$ satisfies the interpolation condition

$$\phi_k(x_{k-1}-x_k)=f(x_{k-1}) \tag{2.8}$$

instead of (2.4) and determines the following value for the scaling parameter

$$\gamma_k=\frac{2}{y_k^T s_k}(f(x_k)-f(x_{k+1})+s_k^T g_{k+1}). \tag{2.9}$$

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 [39] showed that the scaled BFGS update (2.1) with γ_k given by (2.9) 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 γ_k. In these cases Yuan restricted γ_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 [31]). Another scaled BFGS method was introduced by Cheng and Li [31]. In this update the scaling parameter γ_k in (2.1) is computed as

$$\gamma_k=\frac{y_k^T s_k}{\|y_k\|^2}, \tag{2.10}$$

obtained as solution of the problem: $\min\|s_k-\gamma_k y_k\|^2$. Observe that (2.10) is exactly one of the spectral stepsizes introduced by Barzilai and Borwein [32]. Therefore, the scaled BFGS method given by (2.1) with γ_k given by (2.10) 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 [40], Cheng and Li [31] present the computational evidence that their spectral scaled BFGS algorithm is top performer versus the standard BFGS and versus the scaled BFGS algorithms by Al-Baali [25], Yuan [28] and Zhang and Xu [33].

3) *Scaled BFGS with diagonal preconditioning and conjugacy condition* (Andrei [34]). Andrei [34] introduced another scaled BFGS update given by (2.1), in which the scaling parameter γ_k is computed in an adaptive manner as:

$$\gamma_k = \min\left\{\frac{y_k^T s_k}{\|y_k\|^2 + \beta_k}, 1\right\}, \tag{2.11}$$

where $\beta_k > 0$ for all $k = 0,1,\ldots$. Since under the Wolfe line search conditions (1.5) and (1.6) $y_k^T s_k > 0$ for all $k = 0,1,\ldots$, it follows that γ_k given by (2.11) is bounded away from zero, i.e. $0 < \gamma_k \leq 1$. It is proved that if γ_k is selected as in (2.11), where $\beta_k > 0$ for all $k = 0,1,\ldots$, then the large eigenvalues of B_{k+1} given by (2.1) are shifted to the left [34]. Intensive numerical experiments showed that this scaled BFGS algorithm with $\beta_k = \left|s_k^T g_{k+1}\right|$ 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 [26, 27], Yuan [28] and Cheng and Li [31]. The theoretical justification of this selection of the parameter γ_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, γ_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 [41] 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 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. γ_k can be selected to minimize a combination of these two conditions:

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

4) *Scaling the first two terms of the BFGS update with a parameter* (Oren and Luenberger [21] and Nocedal and Yuan [22]). This scaled BFGS update is defined as:

$$B_{k+1} = \delta_k \left[B_k - \frac{B_k s_k s_k^T B_k}{s_k^T B_k s_k} \right] + \frac{y_k y_k^T}{y_k^T s_k}, \tag{2.12}$$

where δ_k is a positive parameter. Concerning the selection of δ_k in (2.12) Oren and Luenberger [21] 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 [22] 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.

5) *Scaling the last terms of the BFGS update with two parameters* (Liao [29]). In another avenue of research, Liao [29] introduced the two parameter modified (scaled) BFGS method:

$$B_{k+1} = B_k - \delta_k \frac{B_k s_k s_k^T B_k}{s_k^T B_k s_k} + \gamma_k \frac{y_k y_k^T}{y_k^T s_k} \tag{2.13}$$

and proved that this scaled BFGS method with two positive parameters corrects the large eigenvalues better than the standard BFGS method given by (1.4) does. In other words, it has been proved that this scaled BFGS method has a strong self-correcting property with respect to the determinant [29]. 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:

$$(\delta_k, \gamma_k) = \begin{cases} \left(\dfrac{s_k^T B_k s_k}{s_k^T B_k s_k + y_k^T s_k}, \dfrac{y_k^T s_k}{s_k^T B_k s_k + y_k^T s_k} \right), & \text{if } \dfrac{s_k^T B_k s_k}{s_k^T B_k s_k + y_k^T s_k} \geq \tau_k, \\ (\tau_k, 1), & \text{otherwise,} \end{cases} \tag{2.14}$$

where $0 < \tau_k < 1$. Liao [29] proposed $\tau_k = \exp(-1/k^2)$. Using a tool given by Byrd and Nocedal [12], Liao proved that the scaled BFGS method given by (2.13)-(2.14) 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 this is competitive with the standard BFGS method and it corrects large eigenvalues better than the standard BFGS method.

3. A Two Parameter Scaled BFGS Update and the TPSBFGS Agorithm

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 (1.4). 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 (1.4) we get:

$$tr(B_{k+1}) = tr(B_k) - \frac{\|B_k s_k\|^2}{s_k^T B_k s_k} + \frac{\|y_k\|^2}{y_k^T s_k}. \tag{3.1}$$

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 [42])

$$\det(I + u_1 u_2^T + u_3 u_4^T) = (1 + u_1^T u_2)(1 + u_3^T u_4) - (u_1^T u_4)(u_2^T u_3) \tag{3.2}$$

where

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

we obtain:

$$\det(B_{k+1}) = \det(B_k) \frac{y_k^T s_k}{s_k^T B_k s_k}. \tag{3.3}$$

In practical implementations the search direction is computed as

$$d_k = -H_k g_k, \tag{3.4}$$

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 (1.4) we get:

$$H_{k+1} = H_k - \frac{H_k y_k s_k^T + s_k y_k^T H_k}{y_k^T s_k} + \left(1 + \frac{y_k^T H_k y_k}{y_k^T s_k}\right)\frac{s_k s_k^T}{y_k^T s_k}. \tag{3.5}$$

Also, for the stepsize computation, in practical implementations the inexact Wolfe line search conditions (1.5) and (1.6) are used.

As we know, the efficiency of the BFGS method is dependent on the structure of the eigenvalues of the approximation to the Hessian matrix [9]. Powell [19] and Byrd, Liu and Nocedal [43] emphasized that the BFGS method actually suffers more from the large eigenvalues than from the small ones. Observe that the second term on the right hand side of (3.1) 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 on the right hand side of (3.1) 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 (1.4) and this is the main motivation for which we use the scaled BFGS methods. In this paper we scale the first two terms in (1.4) with a positive scaling parameter and the third one with another positive scaling parameter in order to correct the large eigenvalues of B_{k+1}. However, it must be a balance between these eigenvalue shifts, otherwise the Hessian approximation could either approach singularity or become arbitrarily large, thus determining the failure of the method [9].

Motivated by the idea of changing the structure of the eigenvalues of the BFGS approximation to the Hessian matrix, in this paper we propose a *double parameter scaled BFGS method* in which the updating of the approximation Hessian matrix B_{k+1} is computed as:

$$B_{k+1} = \delta_k \left[B_k - \frac{B_k s_k s_k^T B_k}{s_k^T B_k s_k} \right] + \gamma_k \frac{y_k y_k^T}{y_k^T s_k}, \tag{3.6}$$

where δ_k and γ_k are positive parameters. In our scaled BFGS method the parameter δ_k is selected to cluster the eigenvalues of B_{k+1}. On the other hand, γ_k is determined to reduce the large eigenvalues of B_{k+1}, thus obtaining a better distribution of the eigenvalues. It is worth saying that a variant of this scaled BFGS update was considered by Nocedal and Yuan [22], where $\delta_k = y_k^T s_k / s_k^T B_k s_k$ and $\gamma_k = 1$. Using the rank-one Sherman-Morrison-Woodbury update formula twice, from (3.6) we get $H_{k+1} = B_{k+1}^{-1}$, where

$$H_{k+1} = \frac{1}{\delta_k}\left[H_k - \frac{H_k y_k s_k^T + s_k y_k^T H_k}{y_k^T s_k} + \left(\frac{\delta_k}{\gamma_k} + \frac{y_k^T H_k y_k}{y_k^T s_k}\right)\frac{s_k s_k^T}{y_k^T s_k} \right], \tag{3.7}$$

is the approximation to the inverse Hessian.

Proposition 2.1. *If the stepsize α_k is determined by the Wolfe line search (1.5) and (1.6), B_k is positive definite and $\gamma_k > 0$, then B_{k+1} given by (3.6) is also positive definite.*

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

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

with equality if $z = 0$ or $s_k = 0$. On the other hand, by the Wolfe line search (1.5) and (1.6) we have that $y_k^T s_k > 0$. Therefore, using the above inequality we get:

$$z^T B_{k+1} z = \delta_k z^T B_k z - \delta_k \frac{z^T B_k s_k s_k^T B_k z}{s_k^T B_k s_k} + \gamma_k \frac{z^T y_k y_k^T z}{y_k^T s_k}$$

$$= \delta_k z^T B_k z - \delta_k \frac{(z^T B_k s_k)^2}{s_k^T B_k s_k} + \gamma_k \frac{(z^T y_k)^2}{y_k^T s_k} \ge \gamma_k \frac{(z^T y_k)^2}{y_k^T s_k} > 0,$$

for any nonzero z. ■

The above proposition says that B_{k+1} given by (3.6) 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 δ_k. Therefore, (3.6) is well defined if $y_k^T s_k > 0$, which is satisfied if the stepsize is determined by the Wolfe line search conditions (1.5) and (1.6). The corresponding scaled BFGS algorithm can be presented as follows.

Two Parameter Scaled BFGS algorithm – TPSBFGS

1. Initialization. Choose an initial point $x_0 \in R^n$ and an initial positive definite matrix H_0. Choose the constants σ, ρ 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 $\|g_k\| \le \varepsilon$, then stop the iterations. Otherwise, continue with step 3.
3. Compute the stepsize $\alpha_k > 0$ satisfying the Wolfe line search conditions (1.5) and (1.6).
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 δ_k and γ_k.
6. Update the inverse Hessian H_k using (3.7).
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 $k = 0,1,...,$ then the above algorithm is exactly the standard BFGS algorithm. *For different values of the parameters δ_k and γ_k in (3.6) (or (3.7)), different scaled BFGS algorithms are obtained.* The algorithm is very general, very easy to implement, but it is applicable only on solving small and medium unconstrained optimization problems.

To implement the TPSBFGS algorithm, some procedures for δ_k and γ_k in step 5 must be given. A variant of TPSBFGS, we consider later in our numerical experiments, is as follows. Since the scaled BFGS with diagonal preconditioning and conjugacy condition where the scaling parameter γ_k is computed in an adaptive manner as:

$$\gamma_k = \min\left\{\frac{y_k^T s_k}{\|y_k\|^2 + \left|s_k^T g_{k+1}\right|}, 1\right\}, \tag{3.8}$$

is the best one, in a variant of the general TPSBFGS algorithm we suggest that γ_k be computed as in (3.8) for all $k = 0,1,\ldots$. Observe that $0 < \gamma_k < 1$.

For selection of δ_k we propose the following strategy. As we know, the performances of the BFGS method are much improved if the eigenvalues of the iteration matrix (3.6) are clustered (see [44]). From (3.6) observe that

$$tr(B_{k+1}) = \delta_k tr(B_k) - \delta_k \frac{\|B_k s_k\|^2}{s_k^T B_k s_k} + \gamma_k \frac{\|y_k\|^2}{y_k^T s_k}. \tag{3.9}$$

Nocedal [9] proved that the third term on the right hand side of (3.1) is bounded by a positive constant. In our algorithm the third term on the right hand side of (3.9) is reduced by the selection of $\gamma_k < 1$ as in (3.8). Since the trace of a matrix is the sum of its eigenvalues, in our double parameter scaled TPSBFGS algorithm we suggest that the parameter δ_k should be selected in such a way that $tr(B_{k+1})$ given by (3.9) to be equal to n. The idea is to select δ_k such that the eigenvalues of B_{k+1} to be clustered. Therefore, from the equation $tr(B_{k+1}) = n$ we obtain:

$$\delta_k = \frac{n - \gamma_k \dfrac{\|y_k\|^2}{y_k^T s_k}}{n - \dfrac{\|B_k s_k\|^2}{s_k^T B_k s_k}}, \tag{3.10}$$

where γ_k is given by (3.8). A characterization of δ_k is as follows.

Proposition 3.1. *Let δ_k be computed as in (3.10). Then, for any $k = 0,1,\ldots$, δ_k is positive and close to 1.*

Proof Observe that along the iterations $\left|s_k^T g_{k+1}\right| \to 0$. Therefore, $\|y_k\|^2 / (\|y_k\|^2 + \left|s_k^T g_{k+1}\right|)$ is close to 1. On the other hand, B_k is symmetric and positive definite. Therefore, it has real and positive eigenvalues: $\lambda_1, \ldots, \lambda_n$. Since B_k is nonsingular and $tr(B_k) = n$, it follows that for any $i = 1, \ldots, n$, $\lambda_i > 0$ 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 $n >> \|B_k s_k\|^2 / s_k^T B_k s_k$. Therefore,

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

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

$$f(x) = \frac{1}{2}(x - x^*)^T G(x - x^*) + f(x^*). \tag{3.11}$$

using the Newton method $x_{k+1} = x_k - \alpha_k H_k g_k$, where H_k is a positive definite matrix and α_k is a stepsize. When the Newton method with the exact line search is applied to minimize (3.11), then the single-step convergence rate can be expressed as:

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

where $R_k = G^{1/2} H_k G^{1/2}$ and $\kappa(R_k)$ is the condition number of R_k. Observe that for the steepest descent method, $R_k = G$ and the single-step convergence rate is linear, with a rate bounded in term of $\kappa(G)$. Luenberger [45] proved that the quasi-Newton DFP method with exact line search applied to minimize (3.11) might cause $\kappa(R_k) > \kappa(G)$ at some iterations. Therefore, in some cases, the DFP method may be inferior to the steepest descent method. Dixon [13] 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 [31]). The following theorem shows that the algorithm TPSBFGS can avoid such cases. For this we need to introduce the following result of Loewner [46].

Proposition 3.2. *Let $A \in R^{n \times n}$ be a symmetric matrix with eigenvalues $\lambda_n \le \lambda_{n-1} \le \cdots \le \lambda_1$ and let $a \in R^n$ be an arbitrary nonzero vector. Denote the eigenvalues of the matrix $\bar{A} = A - aa^T$ by $\mu_n \le \mu_{n-1} \le \cdots \le \mu_1$. Then, we have $\mu_n \le \lambda_n \le \mu_{n-1} \le \lambda_{n-1} \le \cdots \le \mu_1 \le \lambda_1$.* ■

Theorem 3.1. *If we apply the algorithm TPSBFGS with γ_k and δ_k selected as in (3.8) and (3.10) respectively, with exact line search and $B_0 = I$ to minimize (3.11), then $\kappa(R_k) \le \kappa(G)$, 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 [31] (see also [21]). 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 $k \ge 0$, $\kappa(R_k) \le \kappa(G)$. Now, let us write (3.6) as

$$H_{k+1}^{-1} = \delta_k H_k^{-1} - \delta_k \frac{H_k^{-1} s_k s_k^T H_k^{-1}}{s_k^T H_k^{-1} s_k} + \gamma_k \frac{y_k y_k^T}{y_k^T s_k}. \tag{3.12}$$

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:

$$R_{k+1}^{-1} = \delta_k R_k^{-1} - \delta_k \frac{R_k^{-1} r_k r_k^T R_k^{-1}}{r_k^T R_k^{-1} r_k} + \gamma_k \frac{r_k r_k^T}{r_k^T r_k}. \tag{3.13}$$

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

$$P = \delta_k \left[R_k^{-1} - \frac{R_k^{-1} r_k r_k^T R_k^{-1}}{r_k^T R_k^{-1} r_k} \right]. \tag{3.14}$$

Observe that $Pr_k = 0$. Therefore, the matrix P has zero as its eigenvalue, which corresponds to r_k as eigenvector. Observe that P in (3.14) can be written as:

$$P = \delta_k R_k^{-1} - \frac{\delta_k}{r_k^T R_k^{-1} r_k} (R_k^{-1} r_k)(R_k^{-1} r_k)^T. \tag{3.15}$$

Now, if we denote the eigenvalues of P by $\mu_1 \geq \mu_2 \geq \cdots \geq \mu_n = 0,$ and having in view the structure of P given by (3.15), then by Proposition 3.2 we have

$$\lambda_1 \geq \mu_1 \geq \lambda_2 \geq \mu_2 \geq \cdots \geq \lambda_n \geq \mu_n = 0. \tag{3.16}$$

From (3.13) we have

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

Therefore, since $Pr_k = 0$ we have $R_{k+1}^{-1} r_k = \gamma_k r_k,$ i.e. R_{k+1}^{-1} has γ_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 μ_j for some $j = 1,\ldots,n-1$. Then, we have $R_{k+1}^{-1} w_j = P w_j = \mu_j w_j,\ j = 1,\ldots,n-1.$ Therefore, $\mu_1, \mu_2, \ldots \mu_{n-1}, \gamma_k$ 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

$$\kappa(R_k^{-1}) \leq \kappa(G^{-1}). \tag{3.17}$$

Now, let us consider that h_1 and h_2 are the largest and the smallest eigenvalues of G^{-1} respectively. Then, (3.17) implies that $[\lambda_n, \lambda_1] \subseteq [h_2, h_1]$. Therefore, from (3.16) it follows that $\mu_1, \mu_2, \ldots, \mu_{n-1} \in [h_2, h_1]$.

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, γ_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 3.1 we see that the parameter γ_k is the key parameter in the economy of the TPSBFGS algorithm. However, selected as in (3.10), the importance of the parameter δ_k consists in clustering the eigenvalues of the iteration matrix.

4. Global Convergence of TPSBFGS

Assume that the level set $S = \{x : f(x) \le f(x_0)\}$ is bounded. From the first Wolfe condition (1.5) it follows that the sequence $\{f(x_k)\}$ is nonincreasing and therefore $\lim_{k\to\infty} f(x_k)$ exists. Besides, $x_k \in S$. In order to establish the global convergence of the algorithm TPSBFGS, some useful propositions are firstly proved as follows, where γ_k is computed as in (3.8) and δ_k is determined as in (3.10). Our analysis is based on the same principles as those presented by Andrei [34] (see also Li and Fukushima [37] and by Byrd and Nocedal [12]).

Proposition 4.1. *Let δ_k be computed as in (3.10) for $k = 0,1,\ldots$. Then, there are the positive constants $0 < \delta < \Delta$ such that for any $j = 0,1,\ldots,k$,*

$$\delta < \delta_k \delta_{k-1} \cdots \delta_j < \Delta. \tag{4.1}$$

Proof From Proposition 3.1 it follows that δ_k is close to 1 for any $k = 0,1,\ldots$. 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 $j = 0,1,\ldots,$ is bounded as in (4.1). ■

Proposition 4.2. *Consider the double parameter scaled B_{k+1} given by (3.6), where γ_k and δ_k are computed as in (3.8) and (3.10), respectively. Then*

$$tr(B_{k+1}) \le \Delta tr(B_0) + (\Delta k + 1) \tag{4.2}$$

and

$$\sum_{i=0}^{k} \frac{\|B_i s_i\|^2}{s_i^T B_i s_i} \le \frac{\Delta}{\delta}(tr(B_0) + k) + \frac{1}{\delta}. \tag{4.3}$$

Proof Observe that

$$\begin{aligned}
tr(B_{k+1}) &= \delta_k tr(B_k) - \delta_k \frac{\|B_k s_k\|^2}{s_k^T B_k s_k} + \gamma_k \frac{\|y_k\|^2}{y_k^T s_k} \\
&= \delta_k \left(\delta_{k-1} tr(B_{k-1}) - \delta_{k-1} \frac{\|B_{k-1} s_{k-1}\|^2}{s_{k-1}^T B_{k-1} s_{k-1}} + \gamma_{k-1} \frac{\|y_{k-1}\|^2}{y_{k-1}^T s_{k-1}} \right) - \delta_k \frac{\|B_k s_k\|^2}{s_k^T B_k s_k} + \gamma_k \frac{\|y_k\|^2}{y_k^T s_k} \\
&= \ldots \\
&= \delta_k \delta_{k-1} \cdots \delta_0 tr(B_0) \\
&\quad - \delta_k \delta_{k-1} \cdots \delta_0 \frac{\|B_0 s_0\|^2}{s_0^T B_0 s_0} + \delta_k \delta_{k-1} \cdots \delta_1 \gamma_0 \frac{\|y_0\|^2}{y_0^T s_0} \\
&\quad - \delta_k \delta_{k-1} \cdots \delta_1 \frac{\|B_1 s_1\|^2}{s_1^T B_1 s_1} + \delta_k \delta_{k-1} \cdots \delta_2 \gamma_1 \frac{\|y_1\|^2}{y_1^T s_1} \\
&\quad \cdots \\
&\quad - \delta_k \delta_{k-1} \frac{\|B_{k-1} s_{k-1}\|^2}{s_{k-1}^T B_{k-1} s_{k-1}} + \delta_k \gamma_{k-1} \frac{\|y_{k-1}\|^2}{y_{k-1}^T s_{k-1}} \\
&\quad - \delta_k \frac{\|B_k s_k\|^2}{s_k^T B_k s_k} + \gamma_k \frac{\|y_k\|^2}{y_k^T s_k}.
\end{aligned} \tag{4.4}$$

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

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

Therefore, since by Proposition 4.1 there are the positive constants $0 < \delta < \Delta$ such that for any $j = 0,1,\ldots,k,\ \delta < \delta_k \delta_{k-1} \cdots \delta_j < \Delta,$ it follows that

$$tr(B_{k+1}) \le \Delta tr(B_0) - \sum_{i=0}^{k} \delta \frac{\|B_i s_i\|^2}{s_i^T B_i s_i} + \sum_{k=1}^{k} \Delta + 1 \le \Delta tr(B_0) + \Delta k + 1. \tag{4.5}$$

From (4.5) we get (4.2).
On the other hand, since B_{k+1} is positive definite, $tr(B_{k+1}) > 0.$ Therefore (4.3) is true. ■

Remark 4.1 If $B_0 = I,$ then

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

Observe that the last inequality in (4.5) shows that the largest eigenvalue of B_{k+1} is strictly smaller than $\Delta tr(B_0) + (\Delta k + 1).$ Therefore, the scaled TPSBFGS method with γ_k given by (3.8) and δ_k given by (3.10) 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 4.3. *If for all k, $\gamma_k \ge m$, where $m > 0$ is a constant, and $\delta_k \ge \theta$, where $\theta > 0$ is a constant, then there is a constant $c > 0$ such that for all k sufficiently large:*

$$\prod_{i=0}^{k} \alpha_i \ge c^k. \tag{4.6}$$

Proof Considering the identity (3.2), the determinant of the scaled B_{k+1} given by (3.6) is as follows:

$$\begin{aligned}
\det(B_{k+1}) &= \det\left(\delta_k B_k \left(I - \frac{s_k s_k^T B_k}{s_k^T B_k s_k} + \frac{\gamma_k}{\delta_k} \frac{B_k^{-1} y_k y_k^T}{y_k^T s_k} \right) \right) \\
&= \det(\delta_k B_k) \det\left(I - s_k \frac{(B_k s_k)^T}{s_k^T B_k s_k} + \frac{\gamma_k}{\delta_k} (B_k^{-1} y_k) \frac{y_k^T}{y_k^T s_k} \right) \\
&= \delta_k^n \det(B_k) \frac{\gamma_k}{\delta_k} \frac{y_k^T s_k}{s_k^T B_k s_k}.
\end{aligned} \tag{4.7}$$

Therefore,

$$\begin{aligned}
\det(B_{k+1}) &= \delta_k^{n-1} \gamma_k \frac{y_k^T s_k}{s_k^T B_k s_k} \det(B_k) \\
&= \left(\delta_k^{n-1} \gamma_k \frac{y_k^T s_k}{s_k^T B_k s_k} \right) \left(\delta_{k-1}^{n-1} \gamma_{k-1} \frac{y_{k-1}^T s_{k-1}}{s_{k-1}^T B_{k-1} s_{k-1}} \right) \det(B_{k-1}) \\
&= \left(\delta_k^{n-1} \gamma_k \frac{y_k^T s_k}{s_k^T B_k s_k} \right) \left(\delta_{k-1}^{n-1} \gamma_{k-1} \frac{y_{k-1}^T s_{k-1}}{s_{k-1}^T B_{k-1} s_{k-1}} \right) \cdots \left(\delta_0^{n-1} \gamma_0 \frac{y_0^T s_0}{s_0^T B_0 s_0} \right) \det(B_0)
\end{aligned}$$

$$= \left(\prod_{i=0}^{k} \delta_i^{n-1} \gamma_i \frac{y_i^T s_i}{s_i^T B_i s_i} \right) \det(B_0). \tag{4.8}$$

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

$$\det(B_{k+1}) \ge \det(B_0) \prod_{i=0}^{k} \theta^{n-1} m \frac{1-\rho}{\alpha_i} = \det(B_0) \left(\theta^{n-1} \right)^{k+1} m^{k+1} (1-\rho)^{k+1} \prod_{i=0}^{k} \frac{1}{\alpha_i}. \tag{4.9}$$

Since $\det(B_{k+1}) \le \left(\frac{1}{n} tr(B_{k+1}) \right)^n$, by using Proposition 4.2, we get

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

Therefore,

$$\prod_{i=0}^{k} \alpha_i \ge \frac{\det(B_0) \theta^{(n-1)(k+1)} m^{k+1} (1-\rho)^{k+1}}{\det(B_{k+1})} \ge \frac{\det(B_0) \theta^{(n-1)(k+1)} m^{k+1} (1-\rho)^{k+1}}{\left(\frac{1}{n} \left(\Delta tr(B_0) + \Delta k + 1 \right) \right)^n}. \tag{4.10}$$

When k is sufficiently large, (4.10) implies (4.6). ■

Remark 4.2. If $B_0 = I$, then

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

Theorem 4.1. *Let* $\{x_k\}$ *be generated by the algorithm TPSBFGS. Then*

$$\liminf_{k \to \infty} \|g_k\| = 0. \tag{4.11}$$

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 (1.5) we have $\sum_{k=0}^{\infty} (-s_k^T g_k) < \infty$. Therefore,

$$\begin{aligned} \infty > \sum_{k=0}^{\infty} (-s_k^T g_k) &= \sum_{k=0}^{\infty} \frac{1}{\alpha_k} s_k^T B_k s_k = \sum_{k=0}^{\infty} \frac{\|g_k\|}{\|B_k s_k\|} s_k^T B_k s_k \\ &= \sum_{k=0}^{\infty} \frac{s_k^T B_k s_k}{\|B_k s_k\|} \|g_k\| \frac{\|B_k s_k\|}{\|B_k s_k\|} = \sum_{k=0}^{\infty} \frac{s_k^T B_k s_k}{\|B_k s_k\|} \|g_k\| \frac{\alpha_k \|g_k\|}{\|B_k s_k\|} \\ &= \sum_{k=0}^{\infty} \|g_k\|^2 \alpha_k \frac{s_k^T B_k s_k}{\|B_k s_k\|^2} \ge \Gamma^2 \sum_{k=0}^{\infty} \alpha_k \frac{s_k^T B_k s_k}{\|B_k s_k\|^2}. \end{aligned} \tag{4.12}$$

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:

$$q\left[\prod_{k=k_0+1}^{k_0+q}\alpha_k\frac{s_k^TB_ks_k}{\|B_ks_k\|^2}\right]^{1/q}\le\sum_{k=k_0+1}^{k_0+q}\alpha_k\frac{s_k^TB_ks_k}{\|B_ks_k\|^2}\le\Omega. \tag{4.13}$$

Hence,

$$\left[\prod_{k=k_0+1}^{k_0+q}\alpha_k\right]^{1/q}\le\frac{\Omega}{q}\left[\prod_{k=k_0+1}^{k_0+q}\frac{\|B_ks_k\|^2}{s_k^TB_ks_k}\right]^{1/q}\le\frac{\Omega}{q^2}\sum_{k=k_0+1}^{k_0+q}\frac{\|B_ks_k\|^2}{s_k^TB_ks_k}$$
$$\le\frac{\Omega}{q^2}\sum_{k=0}^{k_0+q}\frac{\|B_ks_k\|^2}{s_k^TB_ks_k}\le\frac{\Omega}{q^2}\left(\frac{\Delta}{\delta}tr(B_0)+\frac{\Delta}{\delta}(k_0+q)+\frac{1}{\delta}\right), \tag{4.14}$$

where the last inequality follows from Proposition 4.2. Now, considering $q\to\infty$, we get a contradiction because of Proposition 4.3 which shows that the left-hand side of the above inequality (4.14) is greater than a positive constant. Therefore, (4.1) is true. ■

Observe that the global convergence of the algorithm TPSBFGS with γ_k given by (3.8) bounded from below and with δ_k given by (3.10) 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 (1.5) and (1.6) and without the convexity assumption on f. Moreover, the above results can be obtained for any positive value for the parameter β_k in (2.11) tending to zero. The superlinear convergence of the scaled BFGS method (3.6) with the scaling parameters γ_k and δ_k given by (3.8) and (3.10) respectively can be proved by using a tool and the results presented by Byrd and Nocedal [12] and Dennis and Moré [7, 8] (see also [37]). 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 scaled BFGS method (3.6) with the scaling parameters given by (3.8) and (3.10), and the line search satisfying the inexact Wolfe line search conditions (1.5) and (1.6), 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 [37]).

Remark 4.3. The scaling factor δ_k in (3.10) is determined only from the equation $tr(B_{k+1})=n$, i.e. using only the trace operator. This is not a limitation. If the stepsizes α_k tend to zero, then, as proved by Byrd, Nocedal and Yuan [11], 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, α_k is bounded away from zero in mean. From (4.8) we see that when $s_i^TB_is_i$ is small relative to $y_i^Ts_i$, for arbitrary 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 $tr(B_{k+1})=n$. ■

5. Numerical Results and Comparisons

In this section we present some numerical results with a Fortran implementation of the scaled BFGS algorithms shown above. For this, the algorithm TPSBFGS is particularized as follows: BFGS (TPSBFGS with $\delta_k = 1$ and $\gamma_k = 1$, i.e. the standard BFGS), BFGSC (TPSBFGS with $\delta_k = 1$ and γ_k given by (2.10), i.e. the scaled BFGS given by Cheng and Li [31]), BFGSB (TPSBFGS with $\delta_k = 1$ and γ_k given by (2.7), i.e. the scaled BFGS proposed by Biggs [26, 27]), BFGSY (TPSBFGS with $\delta_k = 1$ and γ_k given by (2.9), i.e. the scaled BFGS suggested by Yuan [28]), BFGSA (TPSBFGS with $\delta_k = 1$ and γ_k given by (2.11), with $\beta_k = \left|s_k^T g_{k+1}\right|$, i.e. the scaled BFGS proposed by Andrei [34]), BFGSD (TPSBFGS with δ_k and γ_k are given by (3.10) and (3.8), respectively, i.e. the scaled BFGS given by Andrei [47]), NOYA (TPSBFGS 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 [22]) and LIAO (scaled BFGS by Liao [29], given (2.13) and (2.14)).

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 $\|.\|_\infty$ 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 (2.13) and (2.14), 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 (3.7). For the scaled BFGS methods by Biggs [26, 27] and Yuan [28], γ_k given by (2.7) and (2.9) 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 scaled BFGS algorithms. Firstly we consider the BFGSD algorithm, where the scaling parameters γ_k and δ_k are given by (3.8) and (3.10), respectively. Consider the problem:

$$\min f(x) = \sum_{i=1}^{n} (\exp(x_i) - \sqrt{i}\, x_i), \tag{5.1}$$

where $n = 10$ and $x_0 = [1,1,\ldots,1]$. 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 $\lambda_1,\ldots,\lambda_{10}$ of the Hessian approximation given by (3.6); the scaling factors γ_k and δ_k given by (3.8) and (3.10), respectively; as well as the evolution of the elements $\|B_k s_k\|^2$ and $s_k^T B_k s_k$ for $k = 1,\ldots,8$.

An attractive feature of the BFGSD algorithm which we see in Table 1 is that along the iterations, the eigenvalues of the Hessian approximation (3.6) are all positive and clustered. In fact, the Hessian approximation (3.6) 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 (3.6) 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 [48]).

From Table 1 we see that δ_k computed as in (3.10) is close to 1, as proved in Proposition 3.1. 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.

Table 1. Characteristics of the BFGSD algorithm.
Eigenvalues of Hessian approximation (3.6), δ_k, γ_k, $\|B_k s_k\|^2$ and $s_k^T B_k s_k$.

k	1	2	3	4	5	6	7	8
λ_1	**0.8532**	**0.6423**	**0.6414**	**0.5303**	**0.4471**	**0.8606**	**1.5591**	**1.5288**
λ_2	**1.0094**	**1.0227**	**1.0075**	**0.9921**	**1.5161**	**0.9380**	**0.6479**	**1.3073**
λ_3	**1.0094**	**1.0227**	**1.0075**	**0.9921**	**1.0637**	**0.9380**	**1.1881**	**0.6181**
λ_4	**1.0094**	**1.0227**	**1.0075**	**0.9921**	**1.0061**	**0.9380**	**0.8883**	**0.8711**
λ_5	**1.0094**	**1.0227**	**1.0075**	**0.9921**	**0.9987**	**0.9388**	**0.9893**	**0.9821**
λ_6	**1.0094**	**1.0227**	**1.0075**	**0.9921**	**0.9951**	**0.9395**	**0.9496**	**0.9427**
λ_7	**1.0094**	**1.0227**	**1.0075**	**0.9966**	**0.9933**	**0.9463**	**0.9451**	**0.9380**
λ_8	**1.0094**	**1.0227**	**1.0185**	**1.0030**	**0.9933**	**0.9656**	**0.9443**	**0.9375**
λ_9	**1.0094**	**1.0749**	**1.0729**	**1.0667**	**0.9933**	**1.0090**	**0.9442**	**0.9372**
λ_{10}	**1.0713**	**1.1239**	**1.2223**	**1.4431**	**0.9933**	**1.5262**	**0.9442**	**0.9372**
δ_k	**1.0094**	**1.0131**	**0.9851**	**0.9847**	**1.0012**	**0.9443**	**1.0065**	**0.9926**
γ_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**

For comparison in Table 2 we present the eigenvalues evolution of the standard BFGS update (1.4) along the iterations for solving the problem (5.1).

Table 2. Characteristics of the standard BFGS algorithm.
Eigenvalues of Hessian approximation (1.4) along the iterations.

k	1	2	3	4	5	6	7	8	9	10	11
λ_1	**0.9810**	**0.8403**	**0.9514**	**0.9134**	**0.8172**	**0.8052**	**0.8561**	**0.8274**	**0.8758**	**0.9220**	**0.9040**
λ_2	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**0.9984**	**0.9880**	**0.9661**
λ_3	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**
λ_4	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1.0007**
λ_5	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1.0052**	**1.0003**	**1.0578**	**1.3407**
λ_6	**1**	**1**	**1**	**1**	**1.0004**	**1.0007**	**1.0689**	**1.0246**	**1.2735**	**1.4572**	**1.8595**
λ_7	**1**	**1**	**1**	**1.0157**	**1.0049**	**1.2222**	**1.1001**	**1.5493**	**1.6327**	**2.0415**	**2.2640**
λ_8	**1**	**1**	**1.1517**	**1.0256**	**1.4972**	**1.3947**	**1.8751**	**1.8844**	**2.2960**	**2.5386**	**2.5981**
λ_9	**1**	**1.4337**	**1.2816**	**1.8659**	**1.8367**	**2.2604**	**2.5229**	**2.7144**	**2.8109**	**2.8102**	**2.8435**
λ_{10}	**2.2009**	**2.3513**	**2.8357**	**2.8916**	**2.9269**	**2.9325**	**2.9374**	**2.9399**	**2.9477**	**2.9478**	**2.9470**

Since both the BFGS update (1.4), and BFGSD update (3.6) where δ_k and γ_k are given by (3.10) and (3.8) respectively, are positive definite, it follows that their eigenvalues are all positive real numbers. Observe the differences between Table 1 and Table 2. From Table 1 observe that for the BFGSD update (3.6) where δ_k and γ_k are given by (3.10) and (3.8), respectively, the maximum eigenvalue along the iterations is 1.5591 and the minimum eigenvalue is 0.4471. On the other hand, for the BFGS update (1.4) the maximum eigenvalue along the iterations is 2.9478 and the minimum eigenvalue is 0.8052. Let us define the *size of the eigenvalues spectrum* of a positive definite matrix as the difference between the largest and the smallest eigenvalues. Observe that the eigenvalues corresponding to the BFGSD algorithm are more clustered. Indeed, the size of the eigenvalues spectrum corresponding to BFGSD algorithm is 1.1120, and the size of the eigenvalues spectrum of BFGS algorithm is 2.1426. Besides the eigenvalues of the BFGSD algorithm are more clustered, observe that in contrast to the BFGS algorithm the eigenvalues spectrum of BFGSD is shifted to the left, i.e. the scaled BFGSD algorithm corrects the large eigenvalues.

Table 3 presents the number of iterations (*iter*) to get a solution of the problem (5.1), the minimum eigenvalue ($\lambda^{\min}$) along the iterations, the maximum eigenvalue ($\lambda^{\max}$) along the iterations and the size of the eigenvalues spectrum (*size*) corresponding to the BFGS and the scaled BFGS algorithms considered in this study.

Table 3.
Characteristics of the eigenvalues of the BFGS and the scaled BFGS algorithms.

	iter	$\lambda^{\min}$	$\lambda^{\max}$	*Size*	References
BFGS	11	0.8052	2.9478	2.1426	Standard BFGS
BFGSA	8	0.5297	1.7008	1.1711	Andrei [34]
BFGSB	21	0.0106	2.2009	2.1903	Biggs [26, 27]
BFGSC	10	0.6233	1.9009	1.2776	Cheng and Li [31]
BFGSD	8	0.4471	1.5591	1.1120	Andrei [47]
BFGSY	14	0.0169	2.2009	2.1840	Yuan [28]
NOYA	13	0.8364	3.7312	2.8948	Nocedal and Yuan [22]

From Table 3 we see that the smallest size of the eigenvalues spectrum corresponds to BFGSD algorithm given by (3.6) where δ_k and γ_k are given by (3.10) and (3.8) respectively. Close to BFGSD is BFGSA where $\delta_k = 1$ and γ_k is computed as in (2.11) with $\beta_k = \left| s_k^T g_{k+1} \right|$. Immediately in order is BFGSC where $\delta_k = 1$ and γ_k is computed as in (2.10). For these algorithms their spectrum is shifted to the left, thus correcting the large eigenvalues. BFGSB and BFGSY have similar performances. They shift the eigenvalues to the left, but their size of the eigenvalues spectrum is larger than that corresponding to BFGSD, BFGSA and BFGSC. The largest size of the eigenvalues spectrum corresponds to NOYA. In the economy of the scaled BFGS algorithms the parameter γ_k has a crucial role (see the Proposition 2.1 and the Theorem 3.1). In NOYA $\gamma_k = 1$ and this is the reason why in NOYA the eigenvalues are not clustered and not shifted to the left.

In the following, we considered a number of 80 unconstrained optimization test problems of medium size ($n = 100$ variables), described in [49]. 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 $i = 1,\ldots,80,$ respectively. We say that, in the particular problem $i,$ the performance of ALG1 was better than the performance of ALG2 if:

$$\left| f_i^{ALG1} - f_i^{ALG2} \right| < 10^{-3} \tag{5.2}$$

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 (3.4) where H_{k+1} is updated as in (3.7) with $\delta_k = 1$ and the corresponding values of γ_k. For the standard BFGS algorithm the search direction is determined as in (3.4) where the approximation to the inverse Hessian is updated as in (3.5).

Figure 1 presents the Dolan and Moré [50] 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 (5.2) hold.

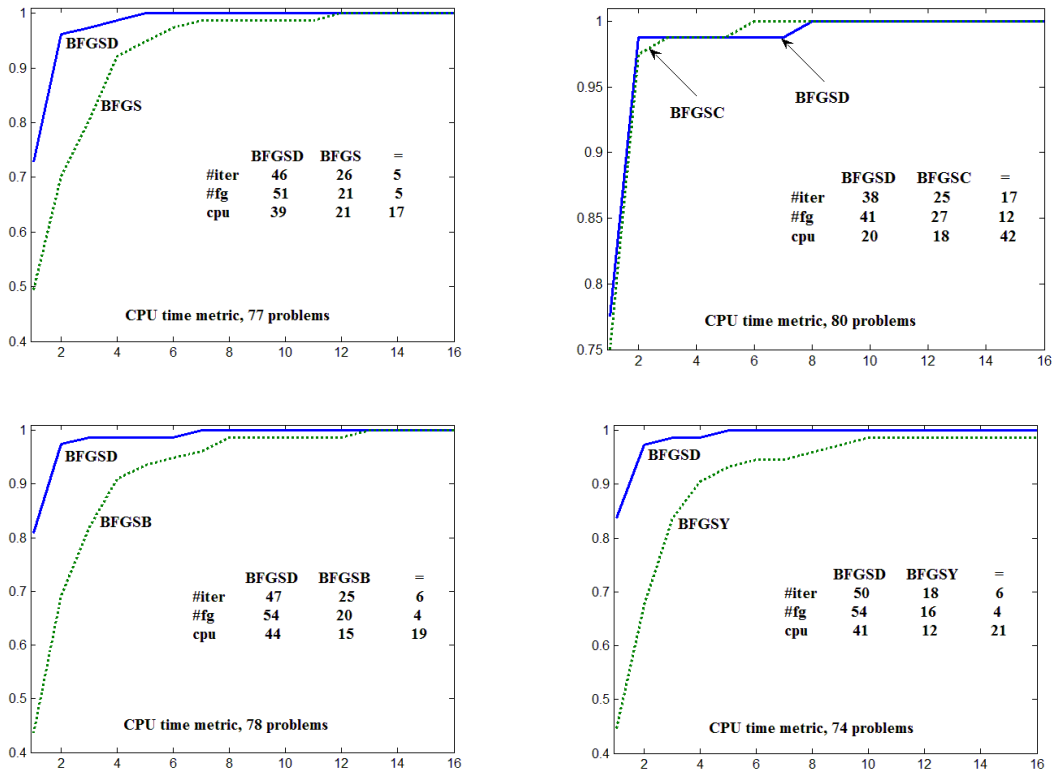

Fig.1. Performance profiles of BFGSD versus BFGS, BFGSC, BFGSB and BFGSY. CPU time metric.

From the performance profiles given in Figure 1 we see that BFGSD is top performer against BFGS, BFGSB, BFGSC and BFGSY algorithms and the differences are significant. 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.

Figure 2 presents the performance profiles of all these 5 scaled 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.

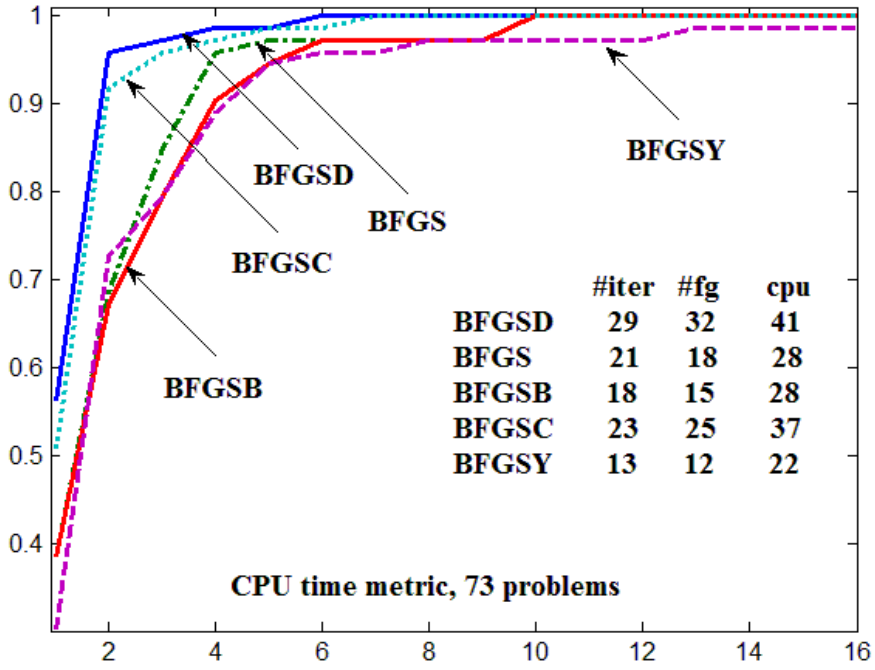

Fig. 2. Performance profile of BFGSD, BFGS, BFGSB, BFGSC and BFGSY. CPU time metric.

In the second set of numerical experiments we compare the double parameter scaled BFGSD algorithm versus the self-scaled BFGS algorithm by Nocedal and Yuan [22], denoted as NOYA, where the approximation of the Hessian B_{k+1} is computed as in (3.6) 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. 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 [22]. 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 to disappointing numerical results. This is consistent with the analysis given by Nocedal and Yuan [22] and Shanno and Phua

[30]. On the other hand, in our study on the double parameter scaled BFGS algorithm BFGSD we emphasize that both parameters γ_k and δ_k are important in the economy of the algorithm: δ_k is computed to cluster the eigenvalues of B_{k+1} and γ_k is responsible for shifting the large eigenvalues to the left. These are the main reasons why BFGSD has better performances than NOYA.

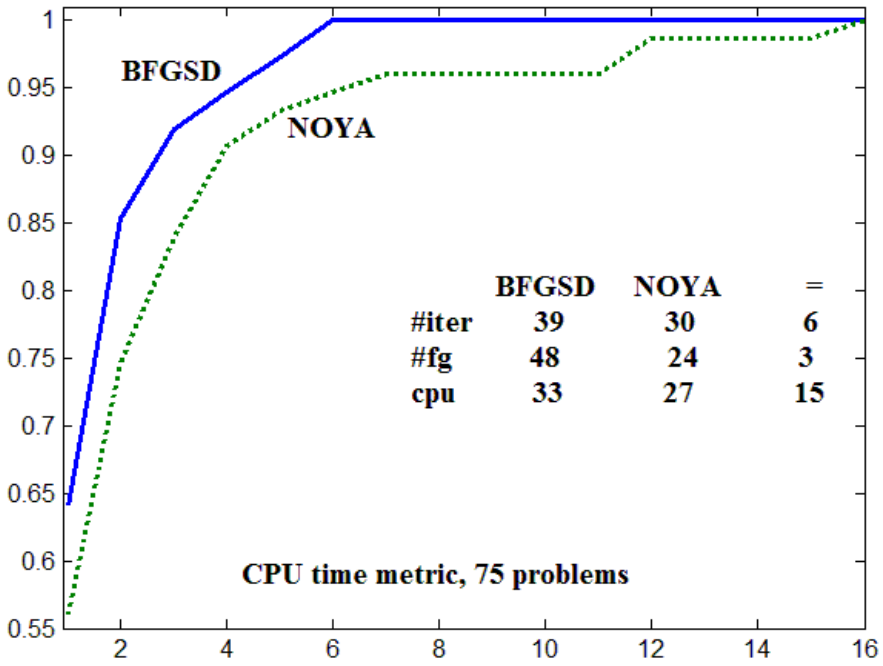

Fig. 3. Performance profile of BFGSD versus NOYA.
CPU time metric.

In the third set of numerical experiments we compare the scaled BFGSD algorithm versus BFGSA. In BFGSA the search direction is determined like in (3.4), where the inverse approximation to the Hessian is computed as in (3.7) with $\delta_k = 1$ and γ_k given by (3.8) [34]. 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 3.1 we proved that δ_k is close to 1. Therefore, in the economy of the BFGSD algorithm, δ_k which scales the first two terms of the BFGS update, is selected as in (3.10) to cluster the eigenvalues of the scaled BFGS matrix in such a way that their sum is 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 [21] 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 contains unity. Besides, in BFGSA the scaling factor γ_k is selected as in (3.8) 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 for BFGSA to be superior to BFGSD.

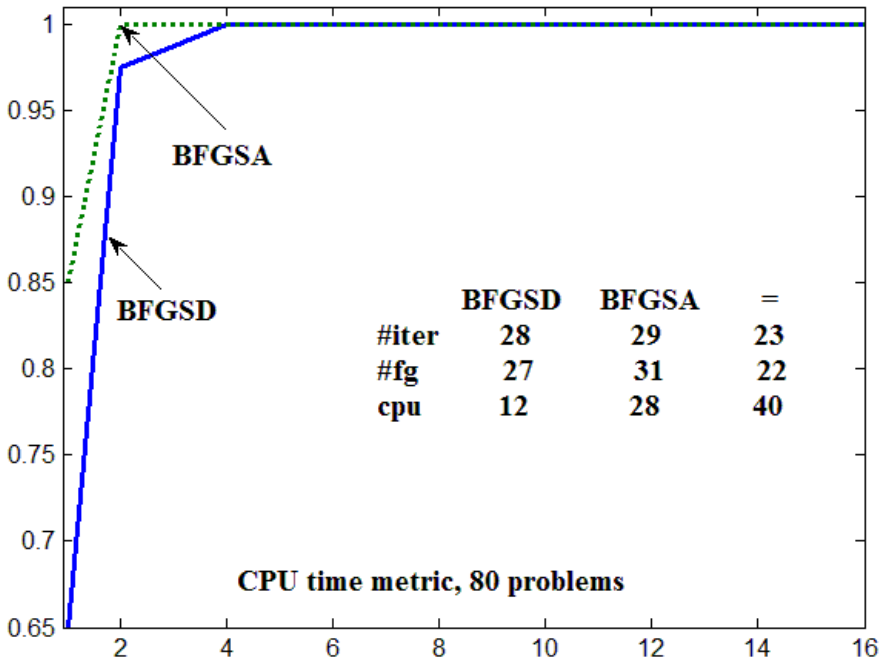

Fig. 4. Performance profile of BFGSD versus BFGSA.
CPU time metric.

In the last set of numerical experiments we compare the double parameter scaled BFGSD algorithm with the scaled BFGS algorithm by Liao [29]. In the Liao algorithm, the Hessian approximation B_{k+1} is computed as in (2.13), where the parameters δ_k and γ_k are computed as in (2.14). 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 τ_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 specified in (2.14). If τ_k is relatively 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 recommended by (2.14). Without drawing too many conclusions from this numerical experiment evidence we note that in both cases the LIAO algorithm finds the local optimal solution. From Figure 5 we see that BFGSD algorithm is top performer versus both variants of LIAO. From (2.13) we get:

$$tr(B_{k+1}) = tr(B_k) - \delta_k \frac{\|B_k s_k\|^2}{s_k^T B_k s_k} + \gamma_k \frac{\|y_k\|^2}{y_k^T s_k}. \tag{5.3}$$

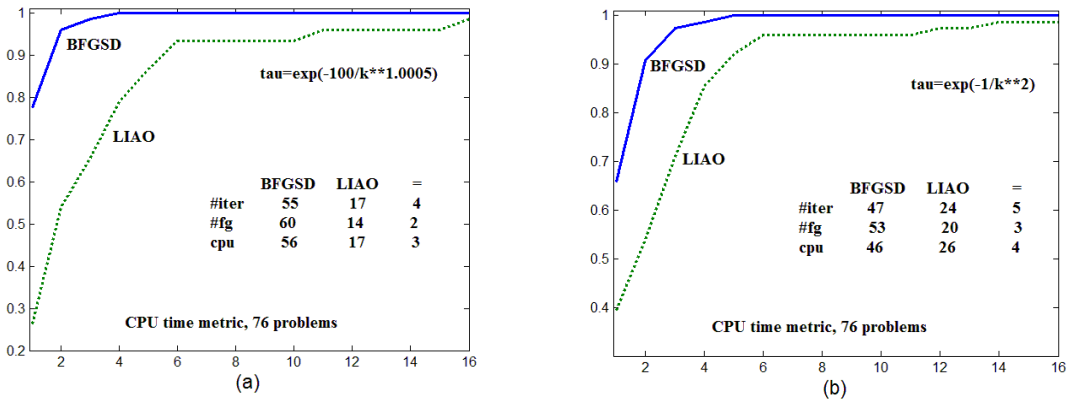

Fig. 5. Performance profile of BFGSD versus LIAO.
CPU time metric.

From (2.14) 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 on the right hand side of (5.3) which shifts the eigenvalues to the left is almost the same as the second term on the right hand side of (3.1), while the third term in (5.3) which shifts the eigenvalues to the right is much smaller than the third term in (3.1). 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 $\gamma_k < 1$ selected as in (3.8), but they are also clustered by a proper selection of δ_k as in (3.10). This is the reason why BFGSD is more efficient and more robust than LIAO (see Fig. 5). In Figure 6 we present a comparison between BFGSD and LIAO with $\tau_k = \exp(-100/k^{1.0005})$, as well as LIAO with $\tau_k = \exp(-1/k^2)$, respectively.

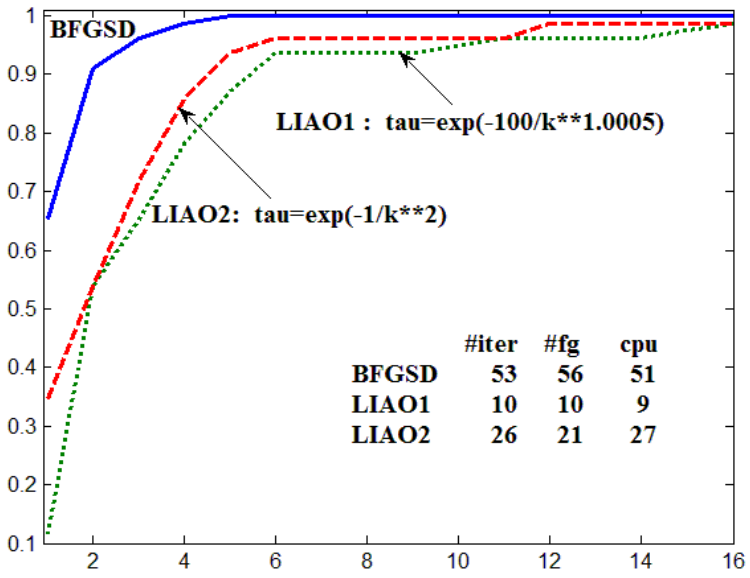

Fig. 6. Performance profiles of BFGSD versus two variants of LIAO.
CPU time metric.

Observe that the best variant of LIAO is that with $\tau_k = \exp(-1/k^2)$ showing that, at least for this set of unconstrained optimization problems considered in this numerical study, the selection of scaling factors δ_k and γ_k in (2.13) as recommended in (2.14) is not a critical one.

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 the 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 with 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 (see also Fig. 6). 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 [29]).

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 $B_{k+1}d_{k+1} = -g_{k+1}$.

6. Conclusions

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 while 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. Preliminary numerical results using a limited number of 80 unconstrained optimization test problems of different structures and complexities show that this double parameter scaled BFGS update is more efficient than the standard BFGS algorithm and also than some other well known scaled BFGS algorithms, including those by Biggs [26, 27], Cheng and Li [31], Liao [29], Nocedal and Yuan [22] and Yuan [28]. 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 [34]). The scaling of this third term 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.

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 an easy task. 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. In the same line of efforts concerning the improving the BFGS method, another interesting idea is to scale the terms of the BFGS update in which y_k is modified as in [51] or in [52], or the scaled BFGS update (3.6) with

modified Wolfe line search used in [53]. Anyway, the BFGS quasi-Newton methods continue to be full of surprises, always having more room for improving their numerical performances.

References

[1] C.G. Broyden, The convergence of a class of double-rank minimization algorithms. I. General considerations, J. Inst. Math. Appl. 6 (1970) 76-90.

[2] R. Fletcher, A new approach to variable metric algorithms, The Computer Journal 13 (1970) 317-322.

[3] D. Goldfarb, A family of variable metric methods derived by variation mean, Mathematics of Computation 23 (1970) 23-26.

[4] D.F. Shanno, Conditioning of quasi-Newton methods for function minimization, Mathematics of Computation 24 (1970) 647-656.

[5] P. Wolfe, Convergence conditions for ascent methods, SIAM Review, 11 (1969) 226-235.

[6] P. Wolfe, Convergence conditions for ascent methods. II: Some corrections, SIAM Review, 13 (1971) 185-188.

[7] J.E. Dennis, J.J. Moré, A characterization of superlinear convergence and its application to quasi-Newton methods, Mathematics of Computation 28 (1974) 549-560.

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

[9] J. Nocedal, Theory of algorithms for unconstrained optimization, Acta Numerica 1 (1992) 199-242.

[10] M.J.D. Powell, 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.

[11] R. Byrd, J. Nocedal, Y. Yuan, Global convergence of a class of quasi-Newton methods on convex problems, SIAM Journal on Numerical Analysis 24 (1987) 1171-1189.

[12] R. Byrd, J. Nocedal, A tool for the analysis of quasi-Newton methods with application to unconstrained minimization, SIAM Journal on Numerical Analysis 26 (1989) 727-739.

[13] L.C.W. Dixon, Variable metric algorithms: necessary and sufficient conditions for identical behavior on nonquadratic functions, Journal of Optimization Theory and Applications 10 (1972) 34-40.

[14] A. Griewank, The global convergence of partitioned BFGS on problems with convex decompositions and Lipschitzian gradients, Mathematical Programming 50 (1991) 141-175.

[15] M.J.D. Powell, On the convergence of the variable metric algorithm, Journal of the Institute of Mathematics and its Applications 7 (1971) 21-36.

[16] W.F. Mascarenhas, The BFGS method with exact line searches fails for non-convex objective functions, Mathematical Programming, Ser. A, 99 (2004) 49-61.

[17] Yu-Hong, Dai, Convergence properties of the BFGS Algorithm, SIAM J. Optim. 13 (2002) 693–701.

[18] R. Fletcher, An overview of unconstrained optimization, in: Algorithms for Continuous Optimization: The State of the Art, E. Spedicato (Ed.), Kluwer Academic Publishers, Boston, (1994) 109-143.

[19] M.J.D. Powell, Updating conjugate directions by the BFGS formula, Mathematical Programming 38 (1987) 693-726.

[20] P.E. Gill, M.W. Leonard, Reduced-Hessian quasi Newton methods for unconstrained optimization, SIAM Journal on Optimization 12 (2001) 209-237.

[21] S.S. Oren, D.G. Luenberger, Self-scaling variable metric (SSVM) algorithms, part I: criteria and sufficient conditions for scaling a class of algorithms, Management Science 20 (1974) 845-862.

[22] J. Nocedal, Y.X. Yuan, Analysis of self-scaling quasi-Newton method, Mathematical Programming 61 (1993) 19-37.

[23] M. Al-Baali, Analysis of a family of self-scaling quasi-Newton methods, Technical Report, Department of Mathematics and Computer Science, United Arab Emirates University, 1993.
[24] M. Al-Baali, Global and superlinear convergence of a class of self-scaling methods with inexact line searches, Comp. Optim. Appl. 9 (1998) 191-203.
[25] M. Al-Baali, Numerical experience with a class of self-scaling quasi-Newton algorithms, Journal of Optimization Theory and Applications 96 (1998) 533-553.
[26] 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 (1971) 315-327.
[27] M.C. Biggs, 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 (1973) 337-338.
[28] Y.X. Yuan, A modified BFGS algorithm for unconstrained optimization, IMA Journal Numerical Analysis 11 (1991) 325-332.
[29] A. Liao, Modifying BFGS method, Operations Research Letters 20 (1997) 171-177.
[30] D.F. Shanno, K.H. Phua, Matrix conditioning and nonlinear optimization, Mathematical Programming 14 (1978) 149-160.
[31] W.Y. Cheng, D.H. Li, Spectral scaling BFGS method, Journal of Optimization Theory and Applications 146 (2010) 305-319.
[32] J. Barzilai, J.M. Borwein, Two-points step size gradient methods, IMA Journal of Numerical Analysis 8 (1988) 141-148.
[33] J.Z. Zhang, C.X. Xu, Properties and numerical performance of quasi-Newton methods with modified quasi-Newton equations, J. Comput. Appl. Math. 139 (2001) 269-278.
[34] N. Andrei, An adaptive scaled BFGS method for unconstrained optimization. Numerical Algorithms, DOI: 10.1007/s11075-017-0321-1.
[35] M. Lalee, J. Nocedal, Automatic column scaling strategies for quasi-Newton methods, Report No. NAM 04, Department of Electrical Engineering and Computer Science, Northwestern University, Evanston, IL, 1991.
[36] D. Siegel, Modifying the BFGS update by a new column scaling technique, Technical Report, Department od Applied Mathematics and Theoretical Physics, Cambridge University, Cambridge, 1991.
[37] D-H. Li, M. Fukushima, A modified BFGS method and its global convergence in nonconvex minimization, Journal of Computational and Applied Mathematics 129 (2001) 15-35.
[38] H.J. Wang, Y.X. Yuan, A quadratic convergence method for one-dimensional optimization, Chinese Journal of Operations Research 11 (1992) 1-10.
[39] M.J.D. Powell, How bad are the BFGS and DFP methods when the objective function is quadratic? Math. Programming, 34 (1986) 34-47.
[40] I. Bongartz, A.R. Conn, N.I.M. Gould, Ph.L. Toint, CUTEr: constrained and unconstrained testing environments, ACM Trans. Math. Software 21, (1995) 123-160.
[41] M.R. Hestenes, E.L. Stiefel, Methods of conjugate gradients for solving linear systems, J. Research Nat. Bur. Standards, 49 (1952) 409-436.
[42] W. Sun, Y.X. Yuan, Optimization theory and methods. Nonlinear programming, Springer Science + Business Media, New York, 2006.
[43] R.H. Byrd, D.C. Liu, J. Nocedal, On the behavior of Broyden's class of quasi-Newton methods, SIAM J. Optim., 2 (1992) 533-557.
[44] N. Andrei, Eigenvalues versus singular values study in conjugate gradient algorithms for large-scale unconstrained optimization, Optimization Methods and Software, 32 (2017) 534-551.
[45] D.G. Luenberger, Introduction to Linear and Nonlinear Programming, Addison-Wesley, Reading, 1973
[46] C. Loewner, Advanced matrix theory, Lecture Notes, Stanford University, Winter (1957)

[47] N. Andrei, A double parameter scaled BFGS method for unconstrained optimization. Preliminary computational results, ICI Technical Report, Bucharest, May 2, 2017.
[48] D.F. Shanno, Conjugate gradient methods with inexact searches, Mathematics of Operations Research, 3, (1978) 244-256.
[49] N. Andrei, An unconstrained optimization test functions collection, Advanced Modeling and Optimization – An Electronic International Journal 10 (2008) 147-161.
[50] E.D. Dolan, J.J. Moré, Benchmarking optimization software with performance profiles, Mathematical Programming 91 (2002) 201-213.
[51] Z. Wei, G. Yu, G. Yuan, Z. Lian, The superlinear convergence of a modified BFGS-type method for unconstrained optimization, Comp. Optim. Appl. 29 (2004) 315-332.
[52] G. Yuan, Z. Wei, Convergence analysis of a modified BFGS method on convex minimizations, Comp. Optim. Appl. 47 (2010) 237-255.
[53] G. Yuan, Z. Wei, X. Lu, Global convergence of BFGS and PRP methods under a modified weak Wolfe-Powell line search, Applied Mathematical Modelling 47 (2017) 811-825.

---oooOooo---