An adaptive scaled BFGS method for unconstrained optimization

Neculai Andrei[1]

March 18, 2017

Abstract. A new adaptive scaled BFGS method for unconstrained optimization is presented. The third term in the standard BFGS update formula is scaled in order to reduce the large eigenvalues of the approximation to the Hessian of the minimizing function. Under the inexact Wolfe line search conditions, the global convergence of the adaptive 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 variant of the scaled BFGS method is more efficient than the standard BFGS update or than some other scaled BFGS methods.

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

Mathematics Subject Classification (2010) 65K. 90C30

1. Introduction

One of the most efficient quasi-Newton methods for solving small and medium-size unconstrained optimization problems is the BFGS method [8, 18, 21, 33]. The theory behind this method and its global convergence are very well established [14, 15]. For convex minimization problems, using the exact line search or some special inexact line search, it has been proved that the BFGS method is globally convergent (see [10, 11, 16, 22, 30]). On the other hand, for non-convex minimization problems, under the exact line search, the BFGS method and other methods in the Broyden class may fail [26]. Also, in [13] Yu-Hong Dai showed that the BFGS method may fail for non-convex functions with line searches that satisfy the Wolfe conditions [38, 39]. However, BFGS has very interesting properties and remains one of the most respectable quasi-Newton methods for unconstrained optimization [19, 27].

As pointed out by Nocedal [27], an interesting property of the BFGS method is its *self-correcting quality*. If the current inverse approximation to the Hessian H_k incorrectly estimates the curvature of the objective function, i.e. if this estimate slows down the iteration, then the Hessian approximation will tend to correct itself within a few steps. Another important property of BFGS explained by Nocedal [27] is that it *better corrects small eigenvalues than large ones*. Powell [32] proved that BFGS with Wolfe inexact line search is globally superlinear convergent for convex problems. On the other hand, Byrd and Nocedal [11] obtained global convergence of BFGS with backtracking line search. Under Wolfe inexact line search, Byrd, Nocedal and Yuan [10] established the global and the superlinear convergence of Broyden's quasi-Newton methods on convex problems (excepting DFP method).

[1] Research Institute for Informatics, Center for Advanced Modeling and Optimization, 8-10 Averescu Avenue, Sector 1, Bucharest, Romania
Academy of Romanian Scientists, 54 Splaiul Independenţei, Sector 5, Bucharest, Romania.

However, intensive numerical experiments showed that the BFGS method may require a large number of iterations or function and gradient evaluations on some problems [20]. The sources of inefficiency of the BFGS method may be caused by a poor initial approximation to the Hessian or 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 [29] scaled the Hessian approximation B_k before updating it, i.e. B_k is replaced 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 [28] 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 $y_k = g_{k+1} - g_k$. They proved that under the Wolfe line search, the corresponding algorithm is globally convergent. 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 impose some conditions on the Broyden family parameter and on the self-scaling factor τ_k. The global convergence and the local superlinear convergence of these 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.

Using different function interpolation conditions Biggs [6, 7] and Yuan [36] obtained some modified BFGS methods and proved their global convergence. The idea of their method is to scale the third term of the BFGS updating formula. The modified BFGS method by Yuan uses both gradient and function values information in one step. Another self-scaling modified BFGS method was suggested by Aiping Liao [25]. In this method the original BFGS updating formula is modified by introducing two positive scaling parameters which correct the eigenvalues of B_k better than the original unscaled BFGS does. Numerical experiments support this claim and indicate that the 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 [11]. A recent spectral scaling BFGS method was proposed by Cheng and Li [12]. 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 [5] 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 [36], Al-Baali [3], Zhang and Xu [37] proved that this spectral scaled BFGS method is clearly more efficient and more robust.

In this paper we introduce an adaptive scaled BFGS method. The idea of this method is to improve the self-correcting property of the BFGS update by scaling the third term of the standard BFGS updating formula. In Section 2 we present the motivation of this new adaptive scaled BFGS method. The scaling factor is computed in an adaptive manner in such a way that the third term of the trace of the scaled BFGS updating formula, which is responsible for shifting the eigenvalues to the right, is reduced. The scaled BFGS updating inherits the positive definiteness of the scaled approximation to the Hessian from the previous iteration, which does not rely on the line search or on the convexity of the minimizing function. In Section 3 the global convergence of

this scaled BFGS method is proved in very general conditions, without assuming the convexity of the minimizing function. Section 4 presents the numerical results obtained with a Fortran implementation of this adaptive scaled BFGS method versus: the standard BFGS update, the modified BFGS method by Yuan [36], the spectral scaled BFGS method by Cheng and Li [12] and the modified BFGS method by Biggs [6, 7]. The numerical results are obtained by solving a set of 80 unconstrained optimization test problems of different structures and complexities [4]. We have the computational evidence that this adaptive scaled BFGS method is much more efficient than the classical BFGS method and than the modified BFGS methods considered in this numerical study.

2. Motivation of scaled BFGS method

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

$$\min f(x), \tag{2.1}$$

where $x \in R^n$. The well known BFGS method for solving (2.1) generates a sequence $\{x_k\}$ computed by the scheme:

$$x_{k+1} = x_k + \alpha_k d_k, \tag{2.2}$$

where d_k is the BFGS search direction obtained as solution of the linear algebraic system

$$B_k d_k = -g_k, \tag{2.3}$$

and g_k is the gradient $\nabla f(x_k)$ of f at x_k. 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 formula:

$$B_{k+1} = B_k - \frac{B_k s_k s_k^T B_k}{s_k^T B_k s_k} + \frac{y_k y_k^T}{y_k^T s_k}, \tag{2.4}$$

where $s_k = x_{k+1} - x_k$, $y_k = g_{k+1} - g_k$, B_0 being symmetric and positive definite. An important property of the BFGS updating formula (2.4), 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 (2.2) is determined by the Wolfe line search conditions:

$$f(x_k + \alpha_k d_k) \le f(x_k) + \sigma \alpha_k g(x_k)^T d_k, \tag{2.5}$$

$$g(x_k + \alpha_k d_k)^T d_k \ge \rho g(x_k)^T d_k, \tag{2.6}$$

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 (2.3) is a descent direction of f at x_k, no matter whether the Hessian is positive definite or not.

It is worth computing the trace and the determinant of the standard B_{k+1} given by (2.4) as important tools in the analysis of the properties and of the convergence of BFGS method. Indeed, by direct computation from (2.4) we get:

$$tr(B_{k+1}) = tr(B_k) - \frac{\|B_k s_k\|^2}{s_k^T B_k s_k} + \frac{\|y_k\|^2}{y_k^T s_k}. \tag{2.7}$$

On the other hand

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

$$= \det(B_k)\det\left(I - s_k \frac{(B_k s_k)^T}{s_k^T B_k s_k} + B_k^{-1} y_k \frac{y_k^T}{y_k^T s_k} \right).$$

Now, applying the identity (see [34])

$$\det(I + u_1 u_2^T + u_3 u_4^T) = (1 + u_1^T u_2)(1 + u_3^T u_4) - (u_1^T u_4)(u_2^T u_3) \tag{2.8}$$

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:

$$\det(B_{k+1}) = \det(B_k) \frac{y_k^T s_k}{s_k^T B_k s_k}. \tag{2.9}$$

We know that the efficiency of the BFGS method is dependent on the structure of the eigenvalues of the approximation to the Hessian matrix [27]. Observe that the second term in (2.7) 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 in (2.7) being positive, it 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 (2.4), and this is the main motivation for which we use the scaled BFGS method. In this paper we scale only the third term in (2.4) for correcting the large eigenvalues of B_{k+1}.

In practical implementations the search direction is computed as

$$d_k = -H_k g_k, \tag{2.10}$$

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 [34] twice, from (2.4) we get:

$$H_{k+1} = H_k - \frac{H_k y_k s_k^T + s_k y_k^T H_k}{y_k^T s_k} + \left(1 + \frac{y_k^T H_k y_k}{y_k^T s_k} \right) \frac{s_k s_k^T}{y_k^T s_k}. \tag{2.11}$$

Also, for the stepsize computation, in practical implementations the inexact Wolfe line search conditions (2.5) and (2.6) are used.

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 scaled BFGS method in which the updating of the approximation Hessian matrix B_k is computed as:

$$B_{k+1} = B_k - \frac{B_k s_k s_k^T B_k}{s_k^T B_k s_k} + \gamma_k \frac{y_k y_k^T}{y_k^T s_k}, \tag{2.12}$$

where γ_k is a positive parameter which is to be determined. Using the rank-one Sherman-Morrison-Woodbury update formula twice, from (2.12) we get:

$$H_{k+1} = H_k - \frac{H_k y_k s_k^T + s_k y_k^T H_k}{y_k^T s_k} + \left(\frac{1}{\gamma_k} + \frac{y_k^T H_k y_k}{y_k^T s_k} \right) \frac{s_k s_k^T}{y_k^T s_k}. \tag{2.13}$$

Proposition 2.1. *If the stepsize α_k is determined by the Wolfe line search (2.5) and (2.6), B_k is positive definite and $\gamma_k > 0$, then B_{k+1} given by (2.12) is also positive definite.*

Proof By the Cauchy-Schwarz inequality, for any $z \neq 0$, we have

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

On the other hand, by the Wolfe line search (2.5) and (2.6) we have that $y_k^T s_k > 0$. Therefore, using the above inequality we get:

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

for any nonzero z. ■

The above proposition shows that B_{k+1} given by (2.12) 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. Therefore, (2.12) is well defined if $y_k^T s_k > 0$, which is satisfied if the stepsize is determined by the Wolfe line search conditions (2.5) and (2.6). With these, the scaled BFGS algorithm can be presented as:

Scaled BFGS algorithm - SBFGS

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 $\|g_k\| \leq \varepsilon$, then stop the iterations. Otherwise, continue with step 3.
3. Compute the stepsize $\alpha_k > 0$ satisfying the Wolfe line search conditions (2.5) and (2.6).
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 factor γ_k.
6. Update the inverse Hessian H_k using (2.13).
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 $\gamma_k = 1$ for all $k = 0,1,...,$ then the above algorithm is exactly the standard BFGS algorithm. For different values of the parameter γ_k in (2.12) (or (2.13)), different scaled BFGS algorithms are obtained. The algorithm is very easy to be implemented, but it is applicable only in solving small and medium unconstrained optimization problems.

Some values for the scaling parameter γ_k in (2.12) have been proposed in literature as follows. Observe that the quasi-Newton step $d_k = -H_k g_k$ is a stationary point of the following problem:

$$\min_{d \in R^n} \phi_k(d) = f(x_k) + g_k^T d + \frac{1}{2} d^T B_k d. \tag{2.14}$$

Since for small d, $\phi_k(d) \approx f(x_k + d)$, it follows that the problem (2.14) is an approximation to the problem (2.1) near the current point x_k. From (2.14) we have that

$$\phi_k(0) = f(x_k), \quad \nabla\phi_k(0) = g(x_k), \tag{2.15}$$

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

$$\nabla\phi_k(x_{k-1} - x_k) = g(x_{k-1}). \tag{2.16}$$

Therefore $\phi_k(x - x_k)$ is a quadratic interpolation of $f(x)$ at x_k satisfying the above conditions (2.15) and (2.16).

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

$$s_{k-1}^T \nabla^2 f(x_k) s_{k-1} = 4 s_{k-1}^T g_k + 2 s_{k-1}^T g_{k+1} - 6(f(x_{k-1}) - f(x_k)). \tag{2.17}$$

Biggs [6, 7] considers the update (2.13) with the value of γ_k chosen in such a way that the new approximate Hessian satisfies the reasonable condition

$$s_{k-1}^T B_k s_{k-1} = 4 s_{k-1}^T g_k + 2 s_{k-1}^T g_{k+1} - 6(f(x_{k-1}) - f(x_k)). \tag{2.18}$$

Therefore, the value of γ_k proposed by Biggs is

$$\gamma_k = \frac{6}{y_k^T s_k}(f(x_k) - f(x_{k+1}) + s_k^T g_{k+1}) - 2. \tag{2.19}$$

For one-dimensional problems Wang and Yuan [35] showed that the scaled BFGS (2.12) with (2.19) and without line search is R-linear convergent.

In the same line of developments, Yuan [36] considers that the approximate function $\phi_k(d)$ satisfies the interpolation condition

$$\phi_k(x_{k-1} - x_k) = f(x_{k-1}) \tag{2.20}$$

instead of (2.16) and determines the following value for the scaling parameter

$$\gamma_k = \frac{2}{y_k^T s_k}(f(x_k) - f(x_{k+1}) + s_k^T g_{k+1}). \tag{2.21}$$

For uniformly convex functions it is easy to prove that there exists a constant $\delta > 0$ such that for all k, $\gamma_k \in [\delta, 2]$. Powell [31] showed that the scaled BFGS method with γ_k given by (2.21) 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 restricts γ_k in the interval $[0.01, 100]$ and proves the global convergence of this variant of the scaled BFGS method.

In another avenue of research Liao [25] introduced a modified (scaled) BFGS method in which both the second and the third terms in (2.4) are scaled with positive scaling parameters:

$$B_{k+1} = B_k - \delta_k \frac{B_k s_k s_k^T B_k}{s_k^T B_k s_k} + \gamma_k \frac{y_k y_k^T}{y_k^T s_k}. \tag{2.22}$$

Liao proved that this scaled BFGS method with two positive parameters corrects the large eigenvalues better than the BFGS method given by (2.4). The parameters scaling the terms in the BFGS update are computed in an adaptive way subject to the values of a positive parameter:

$$(\delta_k, \gamma_k) = \begin{cases} \left(\dfrac{s_k^T B_k s_k}{s_k^T B_k s_k + y_k^T s_k}, \dfrac{y_k^T s_k}{s_k^T B_k s_k + y_k^T s_k} \right), & \text{if } \dfrac{s_k^T B_k s_k}{s_k^T B_k s_k + y_k^T s_k} \geq \tau_k, \\ (\tau_k, 1), & \text{otherwise,} \end{cases} \tag{2.23}$$

where $0 < \tau_k < 1$ (for example $\tau_k = \exp(-1/k^2)$). Using a tool given by Byrd and Nocedal [11], Liao proved that the scaled BFGS (2.22) with the scaling parameters given by (2.23) and using the Wolfe line search generates iterates which converge superlinearly to the optimal solution.

Another scaled BFGS method was introduced by Cheng and Li [12]. In this method the scaling parameter γ_k is computed as

$$\gamma_k = \frac{y_k^T s_k}{\|y_k\|^2}, \tag{2.24}$$

obtained as solution of the problem: $\min\|s_k - \gamma_k y_k\|^2$. Observe that (2.24) is exactly one of the spectral stepsizes introduced by Barzilai and Borwein [5]. Therefore, the scaled BFGS method given by (2.12) with (2.24) 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.

In this paper we introduce another scaled BFGS update given by (2.12), in which the scaling parameter γ_k is computed as:

$$\gamma_k = \min\left\{\frac{y_k^T s_k}{\|y_k\|^2 + \beta_k}, 1\right\}, \tag{2.25}$$

where $\beta_k > 0$ for all $k = 0,1,\ldots$.

Since under the Wolfe line search conditions (2.5) and (2.6) $y_k^T s_k > 0$ for all $k = 0,1,\ldots,$ it follows that γ_k given by (2.25) is bounded away from zero, i.e. $0 < \gamma_k \le 1$.

As we know, the BFGS method actually suffers more from the large eigenvalues than from the small ones (see Powell [32] or Byrd, Liu and Nocedal [9]). Therefore, the motivation behind this adaptive computation of the scaling parameter γ_k as in (2.25) is that if $\gamma_k \le 1$, then the structure of the large eigenvalues of B_{k+1} is corrected by their shifting to the left, as it is proved in the following proposition.

Proposition 2.2. *If γ_k is selected as in (2.25), where $\beta_k > 0$ for all $k = 0,1,\ldots,$ then the large eigenvalues of B_{k+1} given by (2.12) are shifted to the left.*

Proof As we know, the sum of the eigenvalues of matrix B_{k+1} is given by $tr(B_{k+1})$. From (2.12) we have

$$tr(B_{k+1}) = tr(B_k) - \frac{\|B_k s_k\|^2}{s_k^T B_k s_k} + \gamma_k \frac{\|y_k\|^2}{y_k^T s_k}. \tag{2.26}$$

Observe that the second term in (2.26), which is negative, is shifting the eigenvalues of B_{k+1} to the left. On the other hand, the third term in (2.26), which is positive, is shifting the eigenvalues to the right. Now, substituting γ_k from (2.25) in (2.26) we get:

$$tr(B_{k+1}) = tr(B_k) - \frac{\|B_k s_k\|^2}{s_k^T B_k s_k} + \frac{\|y_k\|^2}{\|y_k\|^2 + \beta_k}. \tag{2.27}$$

Since $\|y_k\|^2/(\|y_k\|^2+\beta_k)<1$ for any positive value of β_k, $k=0,1,\ldots,$ it follows that the third term in (2.26) responsible for the large eigenvalues of B_{k+1} is reduced. Therefore, $tr(B_{k+1})$ is reduced, i.e. the large eigenvalues of B_{k+1} are shifted to the left, thus correcting the structure of the eigenvalues. ■

The interpretation of γ_k from (2.24) given by Cheng and Li [12] is that the $\gamma_k I$ is a diagonal preconditioner of $\nabla^2 f(x_{k+1})$, determined in such a way as to reduce the condition number of the Hessian $\nabla^2 f(x_{k+1})$. Cheng and Li suggest that $\gamma_k I$ should be a rough approximation to the inverse of $\nabla^2 f(x_{k+1})$. By minimizing $\|s_k-(\gamma_k I)y_k\|^2$ subject to γ_k, the proposal (2.24) is obtained. However, another interpretation of γ_k given by (2.24) is possible. Observe that from (2.26), a simple way to reduce the value of $tr(B_{k+1})$ is to select γ_k like in (2.24), i.e. exactly the proposal given by Cheng and Li [12], which leads to

$$tr(B_{k+1})=tr(B_k)-\frac{\|B_k s_k\|^2}{s_k^T B_k s_k}+1. \tag{2.28}$$

It is clear that by using (2.25), the reduction of $tr(B_{k+1})$ given by (2.27) is more emphasized than the one given by (2.24).

3. Convergence analysis

Assume that the level set $S=\{x: f(x)\le f(x_0)\}$ is bounded. From the first Wolfe condition (2.5) 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 SBFGS, some useful propositions are proved as follows, where γ_k is computed as in (2.25) and β_k is a positive parameter. Our analysis is based on the same principles as those presented by Li and Fukushima [24] and by Byrd and Nocedal [11].

Proposition 3.1. *Consider the scaled B_{k+1} given by (2.12), where γ_k is computed as in (2.25), then*

$$tr(B_{k+1})\le tr(B_0)+(k+1) \quad \text{and} \quad \sum_{i=0}^{k}\frac{\|B_i s_i\|^2}{s_i^T B_i s_i}<tr(B_0)+(k+1). \tag{3.1}$$

Proof From (2.12) we have:

$$\begin{aligned} tr(B_{k+1}) &= tr(B_k)-\frac{\|B_k s_k\|^2}{s_k^T B_k s_k}+\gamma_k\frac{\|y_k\|^2}{y_k^T s_k} \\ &= tr(B_0)-\sum_{i=0}^{k}\frac{\|B_i s_i\|^2}{s_i^T B_i s_i}+\sum_{i=0}^{k}\gamma_i\frac{\|y_i\|^2}{y_i^T s_i}. \end{aligned} \tag{3.2}$$

But

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

Therefore,

$$tr(B_{k+1}) \le tr(B_0) - \sum_{i=0}^{k} \frac{\|B_i s_i\|^2}{s_i^T B_i s_i} + (k+1) \le tr(B_0) + (k+1). \tag{3.3}$$

Since B_{k+1} is positive definite, $tr(B_{k+1}) > 0$. Therefore (3.1) is true. ■

Remark 3.1. If $B_0 = I$, then

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

Observe that the last inequality in (3.3) shows that the largest eigenvalue of B_{k+1} is strictly smaller than $tr(B_0) + (k+1)$. Therefore, the scaled BFGS method with γ_k given by (2.25) 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 3.2. *If for all k, $\gamma_k \ge m$, where $m > 0$ is a constant, then there is a constant $c > 0$ such that for all k sufficiently large:*

$$\prod_{i=0}^{k} \alpha_i \ge c^k. \tag{3.4}$$

Proof Considering the identity (2.8), the determinant of the scaled B_{k+1} given by (2.12) is as follows:

$$\begin{aligned} \det(B_{k+1}) &= \det\left[B_k \left(I - \frac{s_k s_k^T B_k}{s_k^T B_k s_k} + \gamma_k \frac{B_k^{-1} y_k y_k^T}{y_k^T s_k} \right) \right] \\ &= \det(B_k)\, \gamma_k \frac{y_k^T s_k}{s_k^T B_k s_k} = \det(B_0) \prod_{i=0}^{k} \gamma_i \frac{y_i^T s_i}{s_i^T B_i s_i}. \end{aligned} \tag{3.5}$$

From (2.25) we get:

$$\det(B_{k+1}) = \det(B_0) \prod_{i=0}^{k} \frac{(y_i^T s_i)^2}{s_i^T B_i s_i \left(\|y_i\|^2 + \beta_i \right)}.$$

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

$$\begin{aligned} \det(B_{k+1}) &= \det(B_0) \prod_{i=0}^{k} \frac{y_i^T s_i}{s_i^T B_i s_i} \frac{y_i^T s_i}{\|y_i\|^2 + \beta_i} \\ &\ge \det(B_0) \prod_{i=0}^{k} \frac{1-\rho}{\alpha_i} m = \det(B_0)(1-\rho)^{k+1} m^{k+1} \prod_{i=0}^{k} \frac{1}{\alpha_i}. \end{aligned} \tag{3.6}$$

Since $\det(B_{k+1}) \le \left[\frac{1}{n} tr(B_{k+1}) \right]^n$, using Proposition 3.1 we get

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

Therefore,

$$\prod_{i=0}^{k}\alpha_i \ge \frac{m^{k+1}(1-\rho)^{k+1}\det(B_0)}{\det(B_{k+1})} \ge \frac{m^{k+1}(1-\rho)^{k+1}\det(B_0)}{\left[\frac{1}{n}(tr(B_0)+(k+1))\right]^n}. \tag{3.7}$$

When k is sufficiently large, (3.7) implies (3.4). ■

Remark 3.2. If $B_0 = I$, then

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

Theorem 3.1. *Let* $\{x_k\}$ *be generated by the algorithm, SBFGS. Then*

$$\liminf_{k\to\infty}\|g_k\| = 0. \tag{3.8}$$

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

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

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

$$q\left[\prod_{k=k_0+1}^{k_0+q}\alpha_k\,\frac{s_k^T B_k s_k}{\|B_k s_k\|^2}\right]^{1/q} \le \sum_{k=k_0+1}^{k_0+q}\alpha_k\,\frac{s_k^T B_k s_k}{\|B_k s_k\|^2} \le \Delta.$$

Hence,

$$\begin{aligned}
\left[\prod_{k=k_0+1}^{k_0+q}\alpha_k\right]^{1/q} &\le \frac{\Delta}{q}\left[\prod_{k=k_0+1}^{k_0+q}\frac{\|B_k s_k\|^2}{s_k^T B_k s_k}\right]^{1/q} \le \frac{\Delta}{q^2}\sum_{k=k_0+1}^{k_0+q}\frac{\|B_k s_k\|^2}{s_k^T B_k s_k} \\
&\le \frac{\Delta}{q^2}\sum_{k=0}^{k_0+q}\frac{\|B_k s_k\|^2}{s_k^T B_k s_k} \le \frac{\Delta}{q^2}(tr(B_0)+(k_0+q+1)),
\end{aligned} \tag{3.9}$$

where the last inequality follows from Proposition 3.1. Now, considering $q\to\infty$, we get a contradiction because of Proposition 3.2 which shows that the left-hand side of the above inequality (3.9) is greater than a positive constant. Therefore, (3.8) is true. ■

Observe that the global convergence of the algorithm SBFGS with γ_k given by (2.25) bounded from below is proved in general conditions without the convexity assumption of function f. This is the best result we obtain in very general assumptions that the function f is bounded from below and the line search is based on the Wolfe line search conditions (2.5) and (2.6) and without the convexity assumption on f. Moreover, the above results are obtained for any positive value for the parameter β_k. The superlinear convergence of the scaled BFGS method (2.12) with the scaling parameter γ_k given by (2.25) can be proved by using the tool and the results presented by Byrd and Nocedal [11] and Dennis and Moré [14, 15] (see [24]). If the Hessian matrix $\nabla^2 f(x)$ of the minimizing function f is Lipschitz continuous at the optimal solution x^* of the problem (2.1), then for any positive definite matrix B_0 the modified BFGS method (2.12) with the scaling parameter given by (2.25) and the line search satisfying the inexact Wolfe line search conditions (2.5) and (2.6), generates a sequence $\{x_k\}$ which converges to x^* superlinearly. This result is obtained in 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 [24]).

4. Numerical results

In this section we report some numerical results obtained with an implementation of the scaled BFGS algorithm – SBFGS. The algorithm SBFGS is particularized as follows: BFGSN (SBFGS with γ_k given by (2.25) for different values of $\beta_k > 0$), BFGS (SBFGS with $\gamma_k = 1$, i.e. the standard BFGS), BFGSC (SBFGS with γ_k given by (2.24), i.e. the scaled BFGS given by Cheng and Li [12]), BFGSB (SBFGS with γ_k given by (2.19), i.e. the scaled BFGS proposed by Biggs [6,7]) and BFGSY (SBFGS with γ_k given by (2.21), i.e. the scaled BFGS suggested by Yuan [36]).

We considered a number of 80 unconstrained optimization test problems of medium size ($n = 100$ variables), described in [4]. All the algorithms implement the Wolfe line search conditions with $\sigma = 0.8$ and $\rho = 0.0001$. The iterations are stopped if the inequality $\|g_k\|_\infty \le 10^{-5}$ is satisfied, where $\|.\|_\infty$ is the maximum absolute component of a vector or if the number of iterations exceeds 10^3. In all the algorithms, for all the problems, the initial matrix $H_0 = I$, i.e. the identity matrix. For the scaled BFGS methods by Biggs and Yuan, γ_k given by (2.19) and (2.21) respectively is restricted in the interval [0.01, 100]. 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.

The algorithms we compare in these numerical experiments find local solutions. Therefore, the comparisons of 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:

$$\left| f_i^{ALG1} - f_i^{ALG2} \right| < 10^{-3} \tag{4.1}$$

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 consider $\beta_k = \left| s_k^T g_{k+1} \right|$, i.e. the scaling parameter γ_k in (2.25) is defined as:

$$\gamma_k = \min\left\{ \frac{y_k^T s_k}{\|y_k\|^2 + \left| s_k^T g_{k+1} \right|}, 1 \right\}, \tag{4.2}$$

that is BFGSN is the scaled BFGS algorithm SBFGS with γ_k given by (4.2). A theoretical justification for this selection of β_k in (2.25) as $\beta_k = \left| s_k^T g_{k+1} \right|$ is as follows. Observe that the approximation Hessian B_{k+1} given by (2.12) satisfies the quasi-Newton equation $B_{k+1} s_k = \gamma_k y_k$. If $\gamma_k = 1$, then the classical quasi-Newton equation is obtained. Now, if $\nabla^2 f(x_{k+1})$ is ill-conditioned and $\gamma_k = 1$, then a poor search direction may be obtained. Since the matrix B_{k+1} approximates $\nabla^2 f(x_{k+1})$ along s_k, it follows that larger round errors and numerical instability may appear in the algorithm. To remedy this situation 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. 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 $\|s_k - \gamma_k y_k\|^2$.

But, as we know, if the initial direction d_0 is selected as $d_0 = -g_0$ (i.e. $H_0 = I$) and the objective function to be minimized is a convex quadratic one:

$$f(x) = \frac{1}{2} x^T A x + b^T x + c, \tag{4.3}$$

where $A = A^T$ is positive definite and the exact line searches are used, that is

$$\alpha_k = \arg\min_{\alpha > 0} f(x_k + \alpha d_k), \tag{4.4}$$

then

$$d_i^T A d_j = 0 \tag{4.5}$$

holds for all $i \neq j$. This is called the conjugacy condition. This relation is the original condition used by Hestenes and Stiefel [23] to derive their conjugate gradient algorithms, mainly for solving symmetric positive-definite systems of linear equations. For the general nonlinear twice continuously differentiable function f, by the mean value theorem, there exists some $\xi \in (0,1)$ such that

$$d_{k+1}^T y_k = \alpha_k d_{k+1}^T \nabla^2 f(x_k + \xi \alpha_k d_k) d_k. \tag{4.6}$$

Therefore, is seems reasonable to replace the old conjugacy condition (4.5) from the quadratic case with the following one:

$$d_{k+1}^T y_k = 0, \tag{4.7}$$

in the case of general nonlinear functions. Now, to improve the convergence of the algorithm, the above conjugacy condition can be extended by incorporating the second-order information. In this respect, the quasi-Newton condition also known as the secant equation:

$$H_{k+1} y_k = s_k, \tag{4.8}$$

where H_{k+1} is a symmetric approximation to the inverse Hessian of function f, can be used. Since for the quasi-Newton method the search direction is computed as $d_{k+1} = -H_{k+1} g_{k+1}$, it follows that:

$$d_{k+1}^T y_k = -(H_{k+1} g_{k+1})^T y_k = -g_{k+1}^T (H_{k+1} y_k) = -g_{k+1}^T s_k, \tag{4.9}$$

thus a new conjugacy condition being obtained.

Therefore, we want our algorithm $\gamma_k I$ to be a diagonal preconditioner of $\nabla^2 f(x_{k+1})$ on one side and to minimize the conjugacy condition (4.7) on the other one. Having in view (4.9) it follows that γ_k can be selected to minimize a combination of these two conditions:

$$\|s_k - \gamma_k y_k\|^2 + \gamma_k^2 \left| s_k^T g_{k+1} \right|. \tag{4.10}$$

Notice that in the second term of (4.10) we use γ_k^2 in order to have a positive term. The minimization of (4.10) subject to γ_k gives:

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

exactly as in (4.2) in the BFGSN algorithm. In this way we see that BFGSN is a combination of the scaled BFGS from the quasi-Newton algorithms with the conjugacy condition from the conjugate gradient algorithms.

Figure 1 presents the Dolan and Moré [17] performance profiles of these algorithms for this set of unconstrained optimization problems based on the CPU time metric. For example, when comparing BFGSN versus BFGS (see Figure 1), subject to the number of iterations, we see that BFGSN was better in 47 problems (i.e. it achieved the minimum number of iterations in 47 problems), BFGS was better in 24 problems. Both of them achieved the same number of iterations in 6 problems, etc. Out of 80 problems considered in this set of numerical experiments only for 77 does the criterion (4.1) hold. From the performance profiles given in Figure 1 we see that BFGSN is top performer against all these 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.

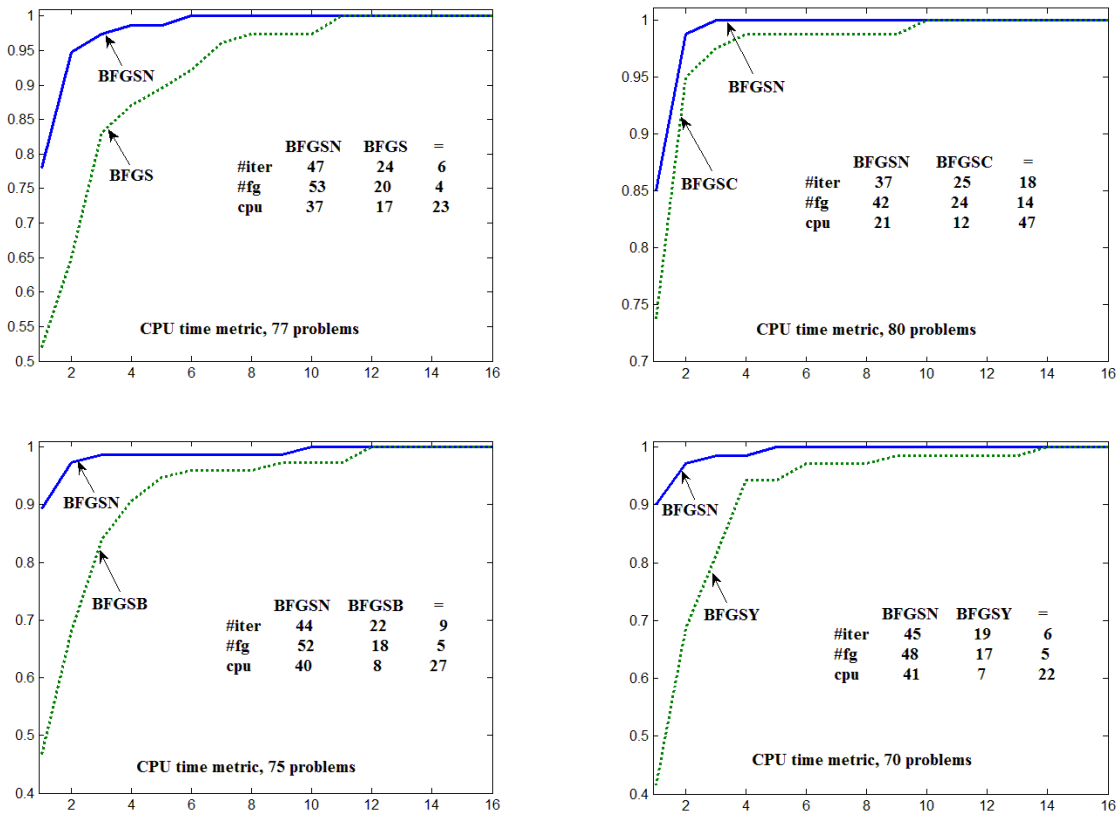

Fig. 1. Performance profiles of BFGSN versus BFGS, BFGSC, BFGSB and BFGSY.

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 BFGSN 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 BFGS methods subject to the CPU computing time metric. We see that BFGSN is top performer, being more efficient than the algorithms considered in this numerical study. From Figure 2 we see that the value of the scaling parameter γ_k given by the adaptive updating formula (4.2) leads us to a scaled BFGS method which is much more efficient and more robust versus the scaled BFGS methods where the value of γ_k is determined by the interpolation conditions given by Biggs [6, 7] or by Yuan [36]. Observe that close to our scaled BFGS method is the one given by Cheng and Li [12]. For this set of numerical experiments, similar performance profiles like in Figure 2 are obtained, subject to the number of iterations or to the number of function and gradient evaluations metrics.

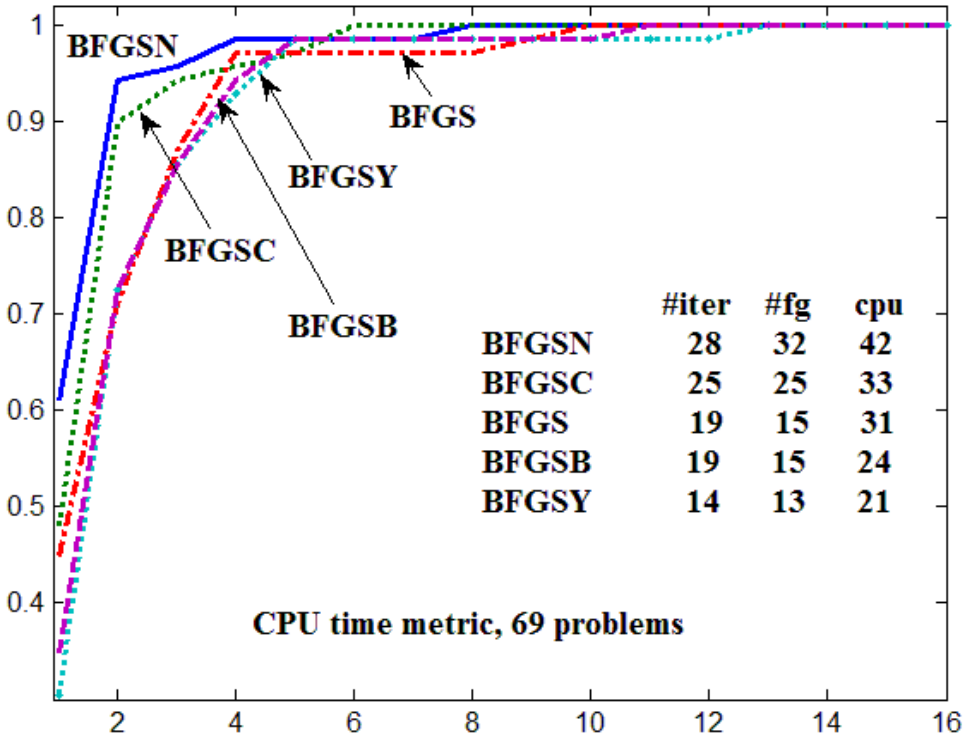

Fig. 2. Performance profiles of BFGSN, BFGSC, BFGS, BFGSB and BFGSY.

By correcting the large eigenvalues and by taking into consideration the minimization of the conjugacy condition, BFGSN is one of the best scaled BFGS algorithms, as proved in the numerical experiments.

In the second set of numerical experiments for the parameter β_k in (2.25) we consider a truncated monotone decreasing evolution. We present the numerical results for two evolutions of β_k in (2.25). In the first one the parameter β_k is computed as

$$\beta_k = \begin{cases} 10^{-k}, & \text{if } k \le 15, \\ 10^{-15}, & \text{if } k > 15, \end{cases} \tag{4.11}$$

and the corresponding algorithm is called BFGSP. In the second case the parameter β_k is computed as

$$\beta_k = \begin{cases} 10^{-k}, & \text{if } k \le 10, \\ 10^{-10}, & \text{if } k > 10, \end{cases} \tag{4.12}$$

and this time the corresponding algorithm is called BFGSQ. In Figure 3 the performance profiles of BFGSN (γ_k given by (4.2)) and BFGS versus these scaled BFGS versions BFGSP and BFGSQ are presented. We see that BFGSN is again top performer versus both BFGSP and BFGSQ. On the other hand, these scaled BFGS versions BFGSP and BFGSQ are much more efficient versus the classical (unscaled) BFGS quasi-Newton updating.

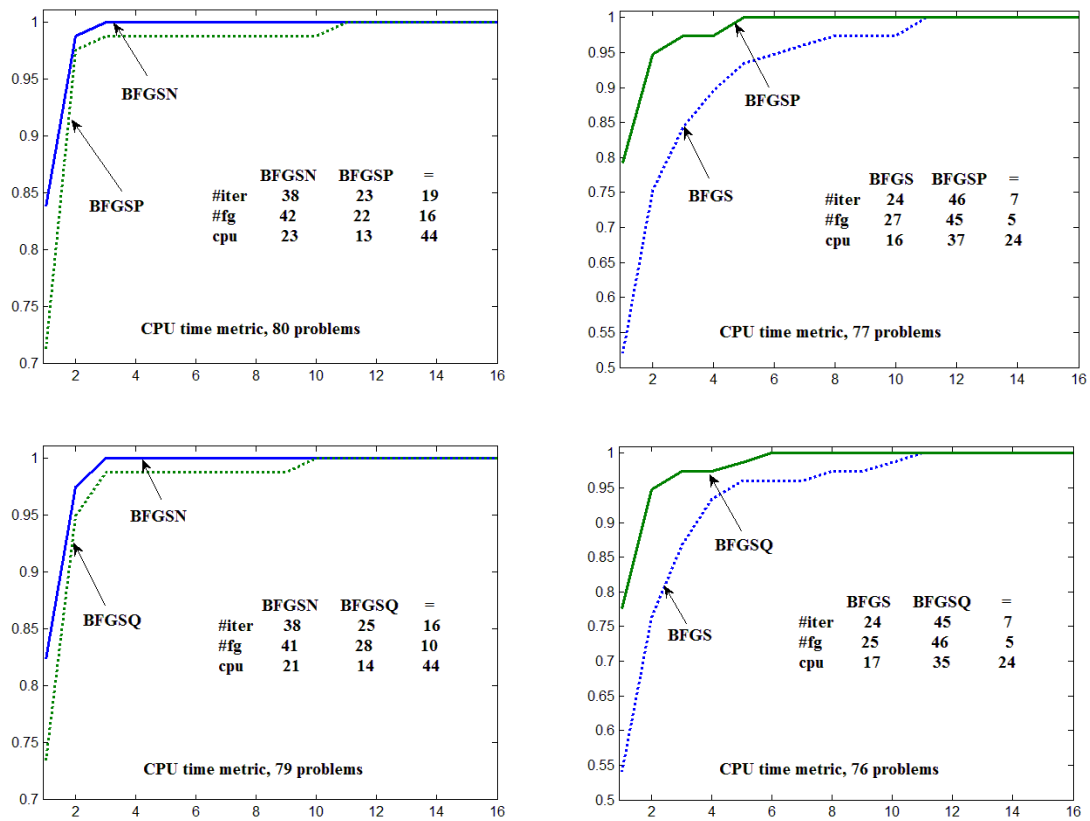

Fig. 3. Performance profiles of BFGSN and BFGS versus BFGSP and BFGSQ.

In the third set of numerical experiments let us consider some small constant values for the parameter γ_k in (2.12). In this case the idea is to see the influence of a constant value of the parameter γ_k on the scaled BFGS performances. Thus, the following three scaled BFGS methods have been considered in our study: BFGSU in which for all $k = 0,1,\ldots$ the value of the parameter γ_k in (2.12) is constant $\gamma_k = 0.1$, BFGSZ in which $\gamma_k = 0.01$ and BFGSS where $\gamma_k = 0.001$. Figure 4 shows the performance profiles of BFGSN and BFGS versus these three variants of the scaled BFGS: BFGSU, BFGSZ and BFGSS. In this figure we see that BFGSN is much more efficient and slightly more robust than BFGSU, BFGSZ and BFGSS. But, on the other hand, all these scaled BFGS variants are more efficient than the classical (unscaled) BFGS quasi-Newton method. This shows once again the importance of scaling the third term in BFGS updating formula (2.12) with a positive scalar, i.e. the importance of correcting the large eigenvalues of the approximation to the Hessian of the minimizing function.

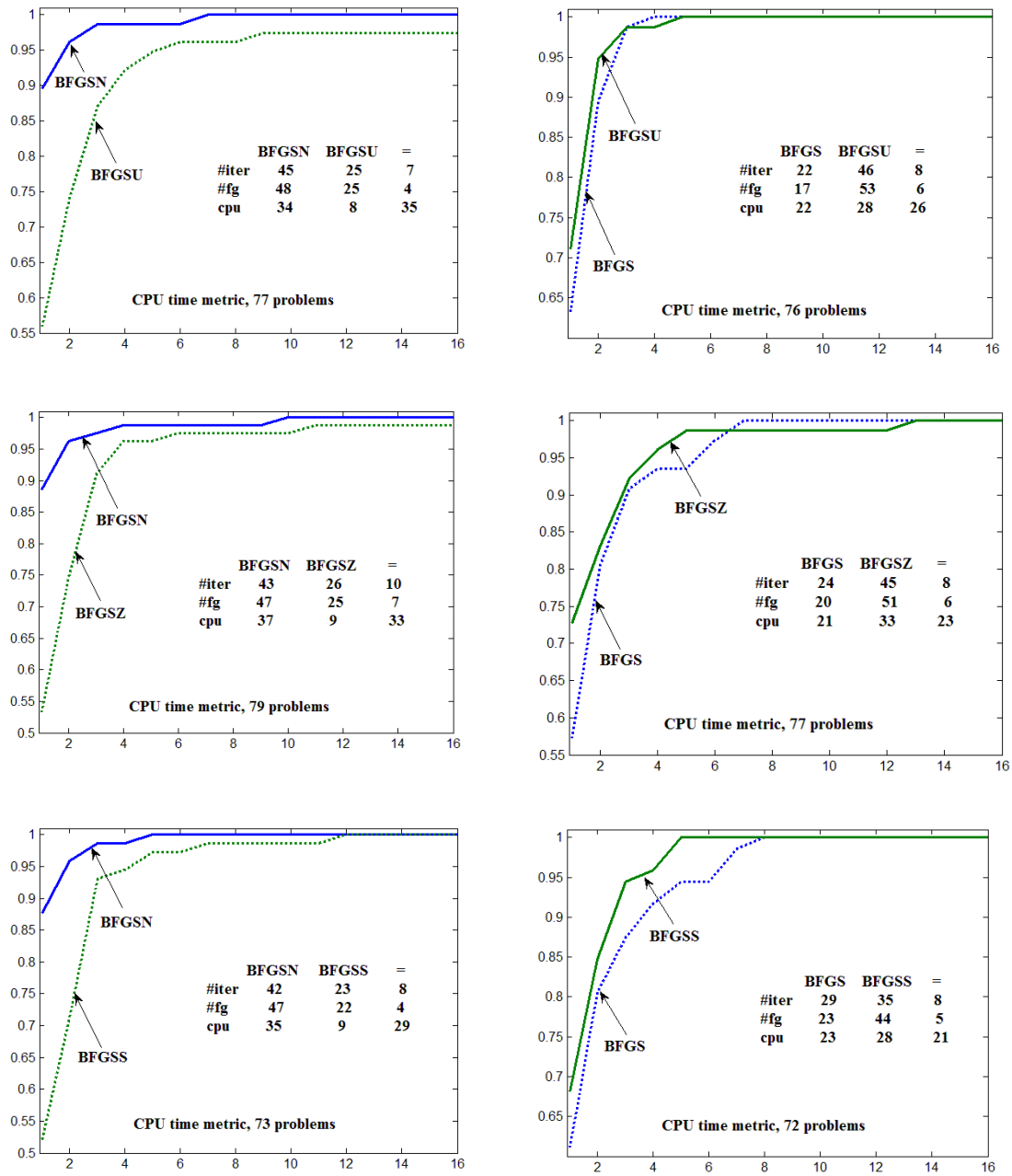

Fig. 4. Performance profiles of BFGSN and BFGS versus BFGSU ($\gamma_k = 0.1$), BFGSZ ($\gamma_k = 0.01$) and BFGSS ($\gamma_k = 0.001$).

From the above numerical experiments we have clear computational evidence that the scaled BFGS update (2.12) with the parameter γ_k as in (2.25) where $\beta_k = \left| s_k^T g_{k+1} \right|$ is the best scaled BFGS quasi-Newton update for solving unconstrained optimization problems. Observe that this algorithm is the best one in comparison to a large diversity of other scaled BFGS updates like: BFGSB, BFGSC, BFGSY, BFGSP, BFGSQ, BFGSU, BFGSZ, and BFGSS.

5. Conclusions

The standard BFGS method has been modified by scaling the third term of its updating formula by a positive parameter. The value of this parameter is computed in an adaptive manner. We show that this scaled BFGS method corrects the large eigenvalues better than the unscaled BFGS method does. We have proved that this adaptive scaled BFGS method with the Wolfe line search is global convergent without assuming the convexity of the minimizing function. Numerical experiments with a limited number of unconstrained minimization test functions, with a medium number of variables, illustrate that this adaptive scaled BFGS method is more efficient than the standard BFGS method or than some other scaled BFGS methods including those by Biggs [6, 7], Yuan [36] and Cheng and Li [12].

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] Barzilai, J., Borwein, J.M.: Two-points step size gradient methods. IMA Journal of Numerical Analysis 8, 141-148 (1988)

[6] 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)

[7] 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)

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

[9] 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)

[10] 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)

[11] 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)

[12] Cheng, W.Y., Li, D.H.: Spectral scaling BFGS method. Journal of Optimization Theory and Applications 146, 305-319 (2010)

[13] Dai, Yu-Hong.: Convergence properties of the BFGS Algorithm. SIAM J. Optim. 13 (3), 693–701 (2002)

[14] 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)

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

[16] 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)

[17] Dolan, E.D., Moré, J.J.: Benchmarking optimization software with performance profiles. Mathematical Programming 91, 201-213 (2002)

[18] Fletcher, R.: A new approach to variable metric algorithms. The Computer Journal 13, 317-322 (1970)
[19] 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)
[20] Gill, P.E., Leonard, M.W.: Reduced-Hessian quasi Newton methods for unconstrained optimization. SIAM Journal on Optimization 12, 209-237 (2001)
[21] Goldfarb, D.: A family of variable metric methods derived by variation mean. Mathematics of Computation 23, 23-26 (1970)
[22] Griewank, A.: The global convergence of partitioned BFGS on problems with convex decompositions and Lipschitzian gradients. Mathematical Programming 50, 141-175 (1991)
[23] Hestenes, M.R., Stiefel, E.L., *Methods of conjugate gradients for solving linear systems*, J. Research Nat. Bur. Standards, 49 (1952), pp.409-436.
[24] 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)
[25] Liao, A.: Modifying BFGS method. Operations Research Letters 20, 171-177 (1997)
[26] Mascarenhas, W.F.: The BFGS method with exact line searches fails for non-convex objective functions. Mathematical Programming, Ser. A, 99, 49-61 (2004)
[27] Nocedal, J.: Theory of algorithms for unconstrained optimization. Acta Numerica 1, 199-242 (1992)
[28] Nocedal, J., Yuan, Y.: Analysis of self-scaling quasi-Newton method. Mathematical Programming 61, 19-37 (1993)
[29] 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)
[30] 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)
[31] Powell, M.J.D.: How bad are the BFGS and DFP methods when the objective function is quadratic? Math. Programming, 34, 34-47 (1986)
[32] Powell, M.J.D.: Updating conjugate directions by the BFGS formula. Mathematical Programming 38, 693-726 (1987)
[33] Shanno, D.F.: Conditioning of quasi-Newton methods for function minimization. Mathematics of Computation 24, 647-656 (1970)
[34] Sun, W., Yuan, Y.X.: Optimization theory and methods. Nonlinear programming. Springer Science + Business Media, New York (2006)
[35] Wang, H.J. Yuan, Y.: A quadratic convergence method for one-dimensional optimization. Chinese Journal of Operations Research 11, 1-10 (1992)
[36] Yuan, Y.: A modified BFGS algorithm for unconstrained optimization. IMA Journal Numerical Analysis 11, 325-332 (1991)
[37] 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)
[38] Wolfe, P.: Convergence conditions for ascent methods. SIAM Review, 11, 226-235 (1969)
[39] Wolfe, P.: Convergence conditions for ascent methods. II: Some corrections. SIAM Review, 13, 185-188 (1971)