BULLETIN *of the*
MALAYSIAN MATHEMATICAL
SCIENCES SOCIETY
http://math.usm.my/bulletin

Bull. Malays. Math. Sci. Soc. (2) **35**(2) *(2012), 247–256*

Scaling on Diagonal Quasi-Newton Update for Large-Scale Unconstrained Optimization

[1]WAH JUNE LEONG, [2]MAHBOUBEH FARID AND [3]MALIK ABU HASSAN

[1,3]Department of Mathematics, Universiti Putra Malaysia, 43400 Serdang, Selangor, Malaysia
[2]Institute for Mathematical Research, Universiti Putra Malaysia, 43400 Serdang, Selangor, Malaysia
[1]leong@math.upm.edu.my, [2]mahboubeh_farid@yahoo.com, [3]malik@math.upm.edu.my

Abstract. Diagonal quasi-Newton (DQN) methods are a class of quasi-Newton methods which alter the standard quasi-Newton updates of approximations to the Hessian or its inverse to diagonal updating matrices. Most often, the updating formulae for this class of methods are derived by the variational approach. A major drawback under this approach is that the derived diagonal matrix may suffer from the loss of positive definiteness and thus it may not be appropriate for use within a descent-gradient algorithm. Previous strategies to overcome this difficulty concentrated on skipping or restarting the non-descent steps. Doing so would abandon the second derivative information that is found on the previous step and consequently, the speed of convergence is usually slower than it would be without these remedies. Hence the present paper intends to propose a simple yet effective remedy to overcome the difficulty that gives arise non-positive-definite updating matrices in the variational based DQN methods. To this end we find that by incorporating an appropriate scaling for the diagonal updating, it improves step-wise convergence while avoiding non-positive definiteness of the updates. Finally, the new DQN method is tested for computational efficiency and stability on numerous test functions, and the numerical results indicate clear superiority over the current methods.

2010 Mathematics Subject Classification: Primary: 65L05; Secondary: 65F10

Keywords and phrases: Quasi-Newton method, diagonal updating, weak quasi-Newton equation, scaling, large-scale optimization.

1. Introduction

Quasi-Newton (QN) methods are considered to be the most efficient methods for solving unconstrained optimization problems of the form:

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

where $x \in \mathscr{R}^n$ and $f \in \mathscr{C}^2$. In QN methods, the basic recursion is analog to the one used in Newton-Raphson method having the form:

$$x_{k+1} = x_k - \alpha_k B_k^{-1} g_k. \tag{1.2}$$

Communicated by Norhashidah Hj. Mohd. Ali.
Received: August 2, 2010; *Revised:* December 28, 2010.

In this recursion, α_k is the stepsize selected to ensure some convergence criteria, while the kth search direction d_k is given by $-B_k^{-1}g_k$ where $g_k = g(x_k) = \bigtriangledown f(x_k)$ is the gradient vector of $f(x)$ and B_k is usually some matrix approximation to the Hessian matrix $G_k = \bigtriangledown^2 f(x_k)$ at x_k, the kth approximation to the solution. The approximations B_k are deferred from the gradients at previous iterations and updated as new gradients become available so as to satisfy the QN equation

$$B_k s_{k-1} = y_{k-1} \tag{1.3}$$

where $s_{k-1} = x_k - x_{k-1}$ and $y_{k-1} = g_k - g_{k-1}$.

This paper is devoted to a class of QN methods that uses some diagonal matrices to approximate the Hessian. The approach underlying such approximation over here was originated by Nazareth [10, 11] where the diagonal approximation is derived based upon the least change weak secant updating strategy of Dennis and Wolkowicz [5] with the added restriction that full matrices are replaced by diagonal matrices. Updating schemes that utilize this approach are then considered by Zhu *et al.* [13] and in particular, the variants that require no linesearch are developed by Leong *et al.* [8] and Hassan *et al.* [7]. In this approach, the variational technique that is employed in the generation of Powell Symmetric Broyden (PSB) and symmetric rank one (SR1) quasi-Newton updates (see, for example Dennis and Schnabel [4]) is utilized to derive the diagonal updating formulae. QN property is incorporated within the variational problem and the resulting updating formulae belong to a class of least change secant updates that are numerically more stable. Like their counterpart of PSB and SR1 updates in the quasi-Newton setting, a major drawback under this approach is that the updated diagonal matrix may suffer from the loss of positive definiteness. Although various measures have been considered by Leong *et al.* [8], Hassan *et al.* [7], and Zhu *et al.*[13] to encounter this limitation, they are generally not very effective (see Section 2 for details). Hence, the main aim of this paper is to propose a simple yet effective remedy for it.

The paper is organized as follows. In Section 2, we formulate and provide short solutions for the variational problems stated in Zhu *et al.* [13], Leong *et al.* [8] and Hassan *et al.* [7] that give the diagonal updating formulae. Section 3 discusses the situation where non-positive-definiteness might occur and propose an effective measure for the difficulty. It follows by computational results in Section 4 to illustrate the merit of our remedy.

2. Diagonal quasi-Newton methods via variational approach

Assume that D_k is positive definite, and let $\{y_k\}$ and $\{s_k\}$ be two sequences of n-vectors such that $y_k^T s_k > 0$ for all k. Because it is usually difficult to satisfy the QN equation, $D_{k+1}s_k = y_k$ with a nonsingular matrix of the diagonal form, one can consider to satisfy it in some directions. By projecting the QN equation (1.3) (also called the secant equation), in a direction υ such that $y_k^T \upsilon \neq 0$ gives

$$s_k^T B_{k+1} \upsilon = y_k^T \upsilon. \tag{2.1}$$

If $\upsilon = s_k$ is chosen, it leads to the so-called weak-secant relation, which was introduced by Dennis and Wolkowicz [5]:

$$s_k^T B_{k+1} s_k = y_k^T s_k. \tag{2.2}$$

Under this weak-secant equation, Zhu *et al.* [13] and Leong *et al.* [8] employ independently, a variational technique that is analogue to the one used to derive the Powell Symmetric Broyden (PSB) quasi-Newton update (see, for example Dennis and Schnabel [4]) for approximating the Hessian matrix diagonally. The resulting update is derived to be the solution of the following variational problem:

$$\begin{aligned} VP1: \quad \min \quad & \frac{1}{2}\|D_{k+1}-D_k\|_F^2 \\ \text{s.t.} \quad & s_k^T D_{k+1} s_k = s_k^T y_k \\ \text{and} \quad & D_{k+1} \text{ is diagonal} \end{aligned}$$

and gives the corresponding solution D_{k+1} as follows:

$$D_{k+1} = D_k + \frac{\left(s_k^T y_k - s_k^T D_k s_k\right)}{tr(E_k^2)} E_k, \tag{2.3}$$

where $E_k = \text{diag}\left(s_{k,1}^2, s_{k,2}^2, \ldots, s_{k,n}^2\right)$, $s_{k,i}$ is the ith component of the vector s_k and tr denotes the trace operator.

Analogously, one can also project the inverse equation, $B_{k+1}^{-1} y_k = s_k$ in the direction $\upsilon = y_k$ to obtain the weak-quasi-Newton equation:

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

Using (2.4), Hassan *et al.* [7] derive the diagonal updating formula for approximating the inverse of Hessian matrix directly as the solution of the following variational problem:

$$\begin{aligned} VP2: \quad \min \quad & \frac{1}{2}\|U_{k+1}-U_k\|_F^2 \\ \text{s.t.} \quad & y_k^T U_{k+1} y_k = y_k^T s_k \\ \text{and} \quad & U_{k+1} \text{ is diagonal} \end{aligned}$$

and leads to the solution U_{k+1}, which is given as

$$U_{k+1} = U_k + \frac{\left(y_k^T s_k - y_k^T U_k y_k\right)}{tr(G_k^2)} G_k, \tag{2.5}$$

where $G_k = \text{diag}\left(y_{k,1}^2, y_{k,2}^2, \ldots, y_{k,n}^2\right)$ and $y_{k,i}$ is the ith component of the vector y_k.

Note that when $s_k^T y_k < s_k^T D_k s_k$ (or $y_k^T s_k < y_k^T U_k y_k$), the resulting D_{k+1} (or U_{k+1}) is not necessarily positive definite. Hence, like their counterpart of PSB update in the quasi-Newton setting, the foregoing update does not preserve positive definiteness and thus it is not appropriate for use within a quasi-Newton-based algorithm.

To address this difficulty, various approaches are considered (see for example, [13], [8], [7]). The first approach is proposed by Zhu *et al.* [13], where they choose to update the square root or Cholesky factor $D^{1/2}$, instead of D. The updating formula for D_{k+1} is then derived as follows:

$$D_{k+1} = \begin{cases} D_k, & \text{if } s_k^T D_k s_k = y_k^T s_k, \\ (I+\mu_k^* E_k)^{-2} D_k, & \text{otherwise,} \end{cases} \tag{2.6}$$

where μ_k^* is the largest solution of the following nonlinear equation:

$$s_k^T (I+\mu E_k)^{-2} D_k s_k = s_k^T y_k. \tag{2.7}$$

As the approach requires the solution of a nonlinear equation at each iteration, when n is large, this strategy would probably cause numerical difficulties. This limits the approach toward solving only small problems. Due to the complexity in computing (2.6), Leong *et al.* [8] proposed to use the following simple updating formula for D_{k+1}:

$$D_{k+1} = \begin{cases} D_k + \frac{\left(s_k^T y_k - s_k^T D_k s_k\right)}{tr(E_k^2)} E_k, & \text{if } D_k + \frac{\left(s_k^T y_k - s_k^T D_k s_k\right)}{tr(E_k^2)} E_k > 0, \\ D_k, & \text{otherwise.} \end{cases} \tag{2.8}$$

The idea is to replace D_{k+1} by D_k, which is supposed to be positive-definite whenever D_{k+1} is not. However, one can see that the resulting D_{k+1} will no longer obey the weak-QN relation if $D_{k+1} = D_k$ is used. This limitation leads to the last approach where Hassan *et al.* [7] proposed to use the following updating formula:

$$D_{k+1} = \begin{cases} D_k + \frac{\left(s_k^T y_k - s_k^T D_k s_k\right)}{tr(E_k^2)} E_k, & \text{if } D_k + \frac{\left(s_k^T y_k - s_k^T D_k s_k\right)}{tr(E_k^2)} E_k > 0, \\ \frac{y_k^T s_k}{y_k^T y_k} I, & \text{otherwise.} \end{cases} \tag{2.9}$$

This updating scheme is equivalent to restart the updating by $(y_k^T s_k / y_k^T y_k)I$ if $D_{k+1} > 0$ is violated. It is interesting to note that $(y_k^T s_k / y_k^T y_k)I$ is precisely the unique matrix that would be obtained from the solution of $VP1$ with the updating matrix is further restricted to a scalar multiple of identity matrix. Hence, the updating matrix of (2.9) will satisfy the weak-QN equation. However when the updating scheme is restarted, information stored during the updating process for D_k may be lost. In fact, both attempts recommended by Leong *et al.* [8] and Hassan *et al.* [7] will abandon the second derivative information that is found on the previous step and consequently the speed of convergence is usually slower than it would be without the skipping/restart.

Motivated by the weaknesses in the existing approaches in handling non-positive-definite updates, we propose a new approach through a scaling strategy (multiplying the approximate Hessian by an appropriate scalar before it is updated) to cater for the weakness. In the following section, we introduce our scaling and present some properties concerning our scaling.

3. Scaling for the diagonal quasi-Newton update

For brevity, let us denote

$$\Lambda_k = \frac{\left(s_k^T y_k - s_k^T D_k s_k\right)}{tr(E_k^2)} E_k. \tag{3.1}$$

and thus, the diagonal updating (2.3) can be expressible as $D_{k+1} = D_k + \Lambda_k$. Firstly, note that the curvature of an objective function, f can be written as

$$s_k^T \bar{G}_k s_k = s_k^T y_k, \tag{3.2}$$

where $\bar{G}_k = \int_0^1 \nabla^2 f(x_k + t s_k) dt$ is the average of Hessian along s_k. Since it is not practical to compute the eigenvalues of $\bar{G}_k$ in each iteration, we can estimate their size relatively to those of D_k on the basis of two useful quantities Q_k and q_k, where Q_k is the Rayleigh quotient of $\bar{G}_k$:

$$Q_k = \frac{s_k^T \bar{G}_k s_k}{s_k^T s_k} = \frac{s_k^T y_k}{s_k^T s_k} \tag{3.3}$$

and q_k is the Rayleigh quotient of D_k:

$$q_k = \frac{s_k^T D_k s_k}{s_k^T s_k} \tag{3.4}$$

with respect to s_k. Thus, an approximation of their relative size may be constructed, on the basis of the scalar

$$\theta_k = \frac{Q_k}{q_k} = \frac{s_k^T y_k}{s_k^T D_k s_k}. \tag{3.5}$$

If $\theta_k > 1$, we can say that the eigenvalues of D_k are relatively small when compared to those of the local Hessian matrix. In addition having $\theta_k > 1$ is also equivalent to have $s_k^T y_k - s_k^T D_k s_k > 0$ and it follows that the corresponding Λ_k is positive (semi-)definite. Hence, one can see that the diagonal updating (2.3) has a self-correcting property in increasing the size of the eigenvalues by adding a positive (semi-)definite Λ_k on D_k and the resulting $D_{k+1} = D_k + \Lambda_k$ will also be positive-definite. Conversely, if $\theta_k < 1$ or equivalently the eigenvalues of D_k are relatively large, we have that $s_k^T y_k - s_k^T D_k s_k < 0$ and subsequently it leads to a negative (semi-)definite Λ_k. The self-correcting property acts by reducing the eigenvalues of D_k through adding a negative definite matrix on D_k of magnitude $|s_k^T y_k - s_k^T D_k s_k|$. However, by reducing the eigenvalues (or diagonal elements) of D_k at the same magnitude may cause some diagonal entries to be over-reduced to become negative. This leads to the present of non-positive definite D_{k+1}. Thus modifying the updating formula (2.3) seems desirable when the size of the eigenvalues of D_k is estimated to be large (i.e. when $\theta_k < 1$). In the following, we try to seek further correction to the large eigenvalues so that the updated D_{k+1} may maintain positive definiteness.

In order to define our strategy, we first describe the so-called scaled diagonal updating formula. The scaled diagonal updating formula is exactly the diagonal updating formula (2.3), except that D_k is replaced by $\sigma_k D_k$:

$$D_{k+1} = \sigma_k D_k + \frac{\left(s_k^T y_k - \sigma_k s_k^T D_k s_k\right)}{tr(E_k^2)} E_k, \tag{3.6}$$

where σ_k is a scaling parameter. If a scaling that is less than or equal to 1 is chosen, Al-Baali [1] showed that a scaled quasi-Newton method will maintain the same convergence property that the original quasi-Newton method has on convex objective functions. Generally, there are two influences that fight against each other in selecting our scaling parameter. First, we observe that scaling may be employed when $\theta_k < 1$. Because choosing a value of $\sigma_k < 1$ decreases the eigenvalues of D_k instantly when D_k is scaled by σ_k. Thus, in this case, the scaled diagonal updating formula has a stronger "reducing" property on large eigenvalues than that of the unscaled diagonal updating formula (2.3). For this purpose, we propose to use the Oren-Luenberger scaling factor [12]:

$$\sigma_k = \theta_k = \frac{s_k^T y_k}{s_k^T D_k s_k}.$$

On the other hand when $\theta_k \geq 1$, scaling is not needed as a scaling that greater than 1 will worsen the situation and may also violate the convergence of the original algorithm. Therefore, we can involve our scaling parameter as

$$\sigma_k = \min\left(\theta_k, 1\right). \tag{3.7}$$

Hence, (3.6) becomes

$$D_{k+1} = \begin{cases} \left(\frac{s_k^T y_k}{s_k^T D_k s_k}\right) D_k, & \text{if } \theta_k < 1, \\ D_k + \frac{\left(s_k^T y_k - s_k^T D_k s_k\right)}{tr(E_k^2)} E_k, & \text{if } \theta_k \geq 1. \end{cases} \tag{3.8}$$

Finally, by combining the feature of scaling to the monotone algorithm of Hassan *et al.* [7] gives our method:

SMDQN Method:

Step 0. Given an initial point x_0 and a positive definite diagonal matrix D_0. Set $k = 0$.
Step 1. If $\|g_k\| \leq \varepsilon$ then stop.
Step 2. If $k = 0$, compute $x_1 = x_0 - g_0/\|g_0\|$. Else if $k \geq 1$, compute $x_{k+1} = x_k - D_k^{-1} g_k$ where D_k is given by (3.8) (with the index $k+1$ is replaced by k) and update D_{k+1}.
Step 3. Let $d_{k,m}$, $d_{k,M}$, $d_{k+1,m}$ and $d_{k+1,M}$ be the smallest and largest diagonal entry of D_k and D_{k+1}, respectively. Check whether $d_{k,m} > d_{k+1,M}/2$ holds. If yes, set $D_{k+1} = \rho I$ where $\rho = \min\{(0.99 d_{k,M})/(2d_{k,m}^2), (s_k^T y_k)/(s_k^T s_k)\}$. Otherwise retain D_{k+1} that is computed in Step 2.
Step 4. Set $k := k+1$ and return to step 1.

The SMDQN method is exactly the method of Hassan *et al.*, except that (3.8) is employed to update D_k. Both of these methods belong to a class of diagonal quasi-Newton methods that do not require linear searches. To get insights into the effect of scaling in algorithmic behaviors, we examine the performance of three methods, include MDQN-I, MDQN-II and SMDQN method in solving the Generalized PSC1 test problem with $n = 100$ [2]:

(1) **MDQN-I** method: SMDQN method with D_k in Step 2 is given by (2.8) (with index $k+1$ be replaced by k).
(2) **MDQN-II** method: SMDQN method with D_k in Step 2 is given by (2.9) (with index $k+1$ be replaced by k).
(3) **SMDQN** method.

We do not consider the strategy of Zhu *et al.* [13] in here as we believe that the strategy is not practical for solving large-scale problems, which are the target group of such methods. The performance of these methods is measured by computing the number, $\log|f(x_k) - f(x^*)|$, where x^* is the minimizer of the function, which the number measures the magnitude of decrease on the objective function in logarithmic scale. Figure 1 illustrates behaviors of the selected algorithms on the minimization. From the figure, one can observe that although the scaling does not alter the trajectory of the DQN direction, it ensures the speed of convergence is higher when compares to that with the skipping/restart (SMDQN method is approximately 90% and 120% faster than that with restarts and skipping, respectively).

4. Convergence analysis

The property of not requiring linear searches is a very important one for much of the effort expended by optimization methods often is spent on these one dimensional minimizations for obtaining the optimal steplength.

In this section, we will consider the convergence of SMDQN method in minimizing a strictly convex quadratic function, f with a constant positive definite Hessian A under some

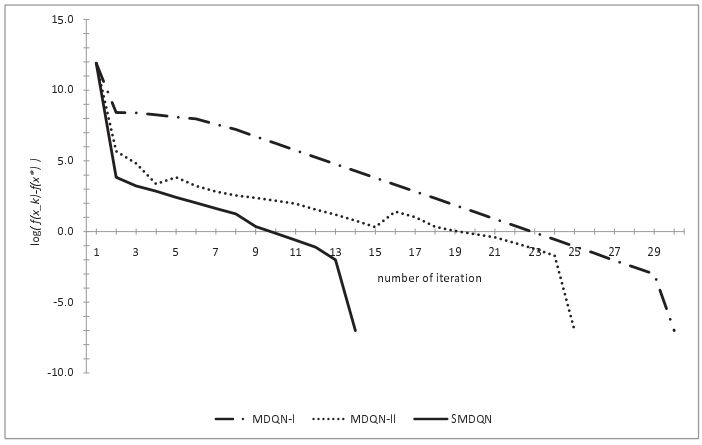

Figure 1. Comparison of the methods: $\log|f(x_k)-f(x^*)|$ vs number of iterations

specific conditions. This is important for it usually also implies convergence for a twice-differentiable nonlinear function within a neighborhood of a local minimum. We shall give the convergence of the SMDQN method as follows:

Theorem 4.1. *Consider the minimization of a strictly convex quadratic function, f with positive definite constant Hessian A. Let $\{x_k\}$ be a sequence generated by the SMDQN method and x^* is a unique minimizer of f. Then either $g_k = 0$ holds for some finite $k \geq 1$ or $\lim_{k\to\infty}\|g_k\| = 0$. Moreover, $\{x_k\}$ converges R−linearly to x^*.*

Proof. By Taylor expansion and the fact $s_k^T A s_k = s_k^T D_{k+1} s_k$, we have

$$\begin{aligned} f(x_k - D_k^{-1}g_k) &= f(x_k) - g_k^T D_k^{-1} g_k + \frac{1}{2} g_k^T D_k^{-1} D_{k+1} D_k^{-1} g_k \\ &= f(x_k) - g_k^T D_k^{-1} D_k D_k^{-1} g_k + \frac{1}{2} g_k^T D_k^{-1} D_{k+1} D_k^{-1} g_k \\ &\leq f(x_k) - \left(d_{k,m} - \frac{d_{k+1,M}}{2} \right) d_{k,M}^{-2} \|g_k\|^2. \end{aligned} \tag{4.1}$$

If $\|g_k\| = 0$, then the first part of the proof is completed. Thus, we assume that $g_k \neq 0$ for all finite k. Note that if the condition

$$d_{k,m} - \frac{d_{k+1,M}}{2} > 0 \tag{4.2}$$

holds, we have that $f(x_{k+1}) \leq f(x_k)$ for all finite k. Else if (4.2) is violated, by Step 3 of SMDQN method, we obtain

$$f(x_k - D_k^{-1}g_k) \leq f(x_k) - \left(d_{k,m} - \frac{\rho d_{k,M}^2}{2} \right) d_{k,M}^{-2} \|g_k\|^2,$$

where ρ is defined as in Step 3 of the SMDQN method. One can see that our choice of ρ will lead to $d_{k,m} - (\rho d_{k,M}^2)/2 > 0$. This implies that in both occasions, $f(x_{k+1}) \leq f(x_k)$

holds for all finite k. Since f is bounded below, we have $f(x_k) - f(x_{k+1}) \to 0$, when $k \to \infty$ and this also implies that $\lim_{k\to\infty} \|g_k\| = 0$.

Furthermore, the strictly convexity of f implies that we can bound $f(x^*)$:

$$f(x) - \frac{1}{2\lambda_m}\|g(x)\|^2 \leq f(x^*) \leq f(x) - \frac{1}{2\lambda_M}\|g(x)\|^2, \tag{4.3}$$

where λ_m and λ_M are the smallest and largest eigenvalues of A, respectively. It follows that $\|g_k\|^2 \geq 2\lambda_m(f(x_k) - f(x^*))$. Thus, (4.1) becomes

$$f(x_{k+1}) - f(x^*) \leq h(f(x_k) - f(x^*)), \tag{4.4}$$

where $h = 1 - c\lambda_m$ with either $c = \left(d_{k,m} - \frac{d_{k+1,M}}{2}\right) d_{k,M}^{-2}$ or $c = d_{k,m} - (\rho d_{k,M}^2)/2$. Note that as $c\lambda_m > 0$ and $f(x_{k+1}) \leq f(x_k)$, we must have $0 < h < 1$ for all k. Therefore the sequence $\{x_k\}$ converges $R-$linearly to x^*. ∎

Note that if (4.2) is violated, this means that some eigenvalues of D_k, in particular those nearby $d_{k,m}$ are relatively too small when compared with those of A. Although the self-correcting property of the updating formula can augment the eigenvalues of D_k to give D_{k+1}, the situation of under-augmented for the eigenvalues that close to $d_{k,m}$ might occur and hence, causes nonmonotone in $\{f(x_k)\}$.

5. Numerical results

To further illustrate the capability of the methods, we solve a set of 30 standard unconstrained optimization problems available in CUTE [3], Moré *et al.* [9] and Andrei [2] with dimension varying from 10 up to 10000.

Table 1. Test problem and its dimension

Test functions (Dimensions)
Freudenstein and Roth, Extended Trigonometric, Extended Beale, Raydan 2, Diagonal 5, Extended Himmelblau, Generalized Rosenbrock, Extended PSC1 Generalized PSC1, Hager, Generalized Tridiagonal 1, Extended Three Exponential Terms, Generalized Tridiagonal 2, Extended Block Diagonal BD1, Quadratic QF2, Extended Tridiagonal 2 Penalty 1, Penalty 2, Full Hessian FH2, EG2, Raydan 1, Diagonal 1, Diagonal 2, Broyden Tridiagonal ($n = 10, 100, 1000, 10000$)
Diagonal 4, Perturbed Quadratic, Diagonal 3, Almost Perturbed Quadratic, Tridiagonal Perturbed Quadratic ($n = 10, 100, 1000$)

The full description of these test problems can be found in [2]. All algorithms are coded in MATLAB 7.0 and are executed on a workstation with dual-processors. All runs are terminated when $\|g_k\| \leq 10^{-5}$. The routine is also forced to stop when the number of iteration exceeds 1000 or CPU time exceeds 10^4 seconds. The performances of these methods, relative to iteration and CPU time, are given in in Figures 2 and 3 using the profiling of Dolan and Morè [6]. We observed from the results that the SMDQN algorithm obtains an improvement over both MDQN-I and MDQN-II methods with an average of 45% and 20% decreases in number of iterations, respectively.

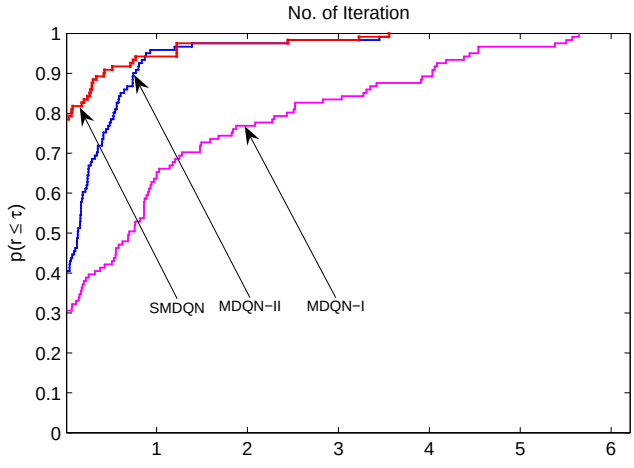

Figure 2. Comparison of the methods: number of iterations

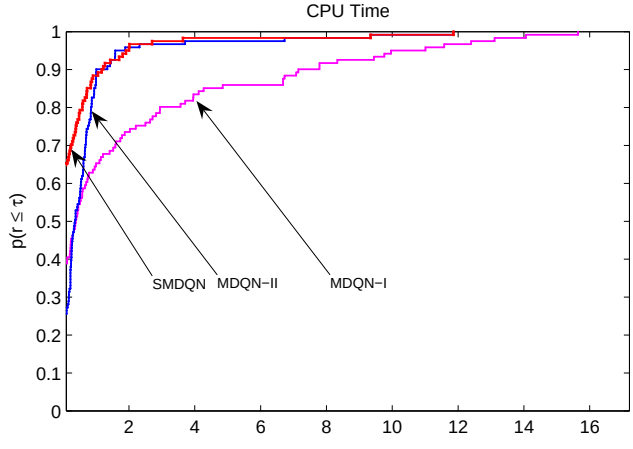

Figure 3. Comparison of the methods: CPU time per iteration

6. Conclusion

This paper suggests a technique to rapidly control large eigenvalues of the diagonal quasi-Newton matrix by scaling the current approximation before updating it. This leads to a simple way in preserving positive definiteness of a DQN updating. The usefulness of our scaling approach within the diagonal quasi-Newton updating, when computational cost is at premium, has been fully demonstrated. Nonetheless, SMDQN algorithm strikes a good compromise for large scale application because it has low time and memory requirements per iteration.

Acknowledgement. The authors are thankful to the anonymous referees for their very valuable comments, which improve the main algorithm significantly. This work was supported by the Malaysian MOHE-FRGS Grant no. 01-11-09-722FR. Part of this work was done while the first author was visiting Academy of Mathematics and Systems Science, Chinese Academy of Sciences, Beijing P. R. China. During the visit, the author was supported by the Joint Chinese Academy of Sciences and Academy of Sciences for the Developing Worlds Fellowship no. 3240157252.

References

[1] M. Al-Baali, On the behaviour of a combined extra-updating/self-scaling BFGS method, *J. Comput. Appl. Math.* **134** (2001), no. 1–2, 269–281.

[2] N. Andrei, An unconstrained optimization test functions collection, *Adv. Model. Optim.* **10** (2008), no. 1, 147–161.

[3] I. Bongartz, A. R. Conn, N. I. M. Gould and Ph. L. Toint, CUTE: constrained and unconstrained testing environment, *ACM Trans. Math. Software* (1995), 123–160.

[4] J. E. Dennis, Jr. and R. B. Schnabel, *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, Prentice Hall Series in Computational Mathematics, Prentice Hall, Englewood Cliffs, NJ, 1983.

[5] J. E. Dennis, Jr. and H. Wolkowicz, Sizing and least-change secant methods, *SIAM J. Numer. Anal.* **30** (1993), no. 5, 1291–1314.

[6] E. D. Dolan and J. J. Moré, Benchmarking optimization software with performance profiles, *Math. Program.* **91** (2002), no. 2, Ser. A, 201–213.

[7] M. A. Hassan, W. J. Leong and M. Farid, A new gradient method via quasi-Cauchy relation which guarantees descent, *J. Comput. Appl. Math.* **230** (2009), no. 1, 300–305.

[8] W. J. Leong, M. A. Hassan and M. Farid, A monotone gradient method via weak secant equation for unconstrained optimization, *Taiwanese J. Math.* **14** (2010), no. 2, 413–423.

[9] J. J. Moré, B. S. Garbow and K. E. Hillstrom, Testing unconstrained optimization software, *ACM Trans. Math. Software* **7** (1981), no. 1, 17–41.

[10] J. L. Nazareth, If quasi-Newton then why not quasi-Cauchy? *SIAG/OPT Views-and-News* 6, 11–14 (1995).

[11] J. L. Nazareth, The quasi-Cauchy method: A stepping stone to derivative-free algorithms, Technical Report No. 95-3, Department of Pure and Applied Mathematics, Washington State University, 1995.

[12] D. F. Shanno and K. H. Phua, Matrix conditioning and nonlinear optimization, Math. Programming **14** (1978), no. 2, 149–160.

[13] M. Zhu, J. L. Nazareth and H. Wolkowicz, The quasi-Cauchy relation and diagonal updating, *SIAM J. Optim.* **9** (1999), no. 4, 1192–1204.