Acta Math. Univ. Comenianae
Vol. LXXVIII, 2(2009), pp. 173–181

QUASI-NEWTON TYPE OF DIAGONAL UPDATING FOR THE L-BFGS METHOD

M. L. SAHARI and R. KHALDI

Abstract. The use of the L-BFGS method is very efficient for the resolution of large scale optimization problems. The techniques to update the diagonal matrix seem to play an important role in the performance of overall method. In this work, we introduce some methods for updating the diagonal matrix derived from quasi--Newton formulas (DFP, BFGS). We compare their performances with the Oren--Spedicato update proposed by Liu and Nocedal (1989) and we get considerable amelioration in the total running time. We also study the convergence of L-BFGS method if we use the BFGS and inverse BFGS update of the diagonal matrix on uniformly convex problems.

1. Introduction

The aim of this paper is to introduce some methods for updating the diagonal matrix obtained from quasi-Newton formulas and then after applying Oren-Spedicato update proposed in [**4**] by Liu and Nocedal to study their performances. We show that the total running time is improved considerably. Using the BFGS and inverse BFGS update of diagonal matrix on uniformly convex problems, we study the convergence of L-BFGS method.

More precisely, we consider the unconstrained optimization problem

$$\min_{x\in\mathbb{R}^n} f(x), \tag{1.1}$$

where f is a real valued function on $\mathbb{R}^n$. We assume throughout that both the gradient $g(x) = \nabla_x f(x)$ and the Hessian matrix $H(x) = \nabla_{xx} f(x)$ of f exist and are continuous. To solve the problem (1.1) one uses an algorithm that generates a sequence of iterates x_k according to

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

where $x_1 \in \mathbb{R}^n$ is given, $d_k \in \mathbb{R}^n$ is the search direction and t_k is a step length which minimizes f along d_k from the point x_k. In this paper we suppose that t_k

Received February 5, 2007; revised May 5, 2009.
2000 *Mathematics Subject Classification.* Primary 65K05, 65K10, 90C30.
Key words and phrases. nonlinear unconstrained optimization; Quasi-Newton; BFGS; L-BFGS; line search.

satisfies the Wolfe conditions (see [**1**], [**6**])

$$f(x_k + t_k d_k) - f(x_k) \leq c_1 t_k g(x_k)^\top d_k, \tag{1.3}$$

$$g(x_k + t_k d_k)^\top d_k \geq c_2 g(x_k) d_k, \tag{1.4}$$

where $0 < c_1 < 1/2$, $c_1 < c_2 < 1$. If n is not large ($n \leq 100$), the BFGS method is very efficient (see [**1**], [**6**]). In this case the direction d_k is defined by

$$d_k = -S_k \cdot g(x_k) \tag{1.5}$$

where S_k is an inverse Hessian approximation updated at every iteration by means of the formula

$$S_{k+1} = V_k^\top S_k V_k + \rho_k \delta_k \delta_k^\top \tag{1.6}$$

where

$$\rho_k = \frac{1}{\gamma_k^\top \delta_k}, \qquad V_k = I - \rho_k \gamma_k \delta_k^\top \tag{1.7}$$

and

$$\delta_k = x_{k+1} - x_k, \qquad \gamma_k = g(x_{k+1}) - g(x_k). \tag{1.8}$$

But if n is very large when the matrix S_k cannot be computed or stored, it is desirable to use limited memory BFGS (L-BFGS) (see [**1**], [**2**], [**3**]). The implementation described by Liu and Nocedal [**4**] is almost identical to that of the BFGS method, the only difference is in the matrix update. Instead of storing the matrix S_k, one stores a certain number, say m of pairs $\{\delta_k, \gamma_k\}$. The product $S_k \cdot g(x_k)$ is obtained by performing a sequence of inner products involving $g(x_k)$ and the m most recent vector pairs $\{\delta_k, \gamma_k\}$. After computing the new iterate, the oldest pair is deleted from the set $\{\delta_k, \gamma_k\}$ and replaced by the newest one. The algorithm therefore always keeps the m most recent pairs $\{\delta_k, \gamma_k\}$ to define the iteration matrix. This strategy is suitable for large scale problems because it has been observed in practice that small values of m (say $m \in \langle 3\,,\, 8\rangle$) give satisfactory results [**4**].

The structure of this paper is the following: In the next section we describe the L-BFGS update process. Section three deals with updating the diagonal matrix, for this we propose several updating methods for the diagonal matrix D_k derived from the quasi-Newton methods [**3**], [**6**].

In section four, we study the convergence, we show that the L-BFGS method with BFGS and inverse BFGS update formulas of the diagonal matrix are globally convergent on uniformly convex problems if the line search is Wolfe type. Section five concerns numerical tests on some problems proposed in [**5**], using FORTRAN 90. Then we give tables that compare the performance of the L-BFGS method with Oren-Spedicato scaling and other updates of the diagonal matrix proposed in this work (DFP, BFGS and Inverse BFGS). We achieve this work with a conclusion and some references.

2. L-BFGS Method

Now, we give a precise description of the L-BFGS update process in more details. Let x_1 be given as the first element of the iteration and suppose that we have stored the m pairs $\{\delta_k, \gamma_k\}$, $k = 1, \ldots, m$. We choose a "basic matrix" $S_k^1 = D_k$ (usually a diagonal with all positive entries) and from (1.6) the S_k update can be written as

$$\tag{2.1} \begin{cases} S_{k+1} = V_k^\top S_k V_k + \rho_k \delta_k \delta_k^\top, & \text{for } 1 \le k \le m \\ S_k^1 = D_k & \\ S_k^{i+1} = V_{k-m+i-1}^\top S_k^i V_{k-m+i-1} & \\ \qquad + \rho_{k-m+i-1} \delta_{k-m+i-1} \delta_{k-m+i-1}^\top, & 1 \le i \le m \\ S_k = S_k^{m+1}, & k \ge m+1. \end{cases}$$

In practice we prefer to use the following more explicit formula

$$S_k = \sum_{i=1}^{i=\widehat{m}+1} \rho_{k-i} \left(\prod_{j=0}^{j=i-1} V_{k-j}^\top \right) \delta_{k-j} \delta_{k-j}^\top \left(\prod_{j=1}^{j=i} V_{k+j-i} \right),$$

with $\widehat{m} = \min\{k-1, m\}$ and

$$\begin{cases} \rho_{k-\widehat{m}-1} = 1, \\ \delta_{k-\widehat{m}-1} \delta_{k-\widehat{m}-1}^\top = D_k, \\ V_k = I. \end{cases}$$

3. Updating the diagonal matrix

The diagonal matrix D_k has to be ensured by the weak quasi-Newton condition

$$\tag{3.1} \gamma_{k-1}^\top S_k \gamma_{k-1} = \gamma_{k-1}^\top \delta_{k-1}.$$

Liu and Nocedal [**4**] recommend the choice of the multiple of the unit matrix

$$\tag{3.2} D_k = \zeta_{k-1} I,$$

where I is the identity matrix and $\zeta_{k-1} = \dfrac{\gamma_{k-1}^\top \delta_{k-1}}{\gamma_{k-1}^\top \gamma_{k-1}}$. This scaling is suggested by Oren and Spedicato [**7**], using only the last pair (δ_i, γ_i). This strategy does not allow a well initialization of the L-BFGS method. In the following we propose several methods of updating the diagonal matrix D_k, derived from the quasi--Newton methods, i.e. DFP and BFGS methods.

3.1. The DFP type of diagonal update

The DFP diagonal update formula is obtained by taking the diagonal of the matrix D_k with the DFP formula. If

$$D_k = \operatorname{diag}\left[D_k^{(1)}, D_k^{(2)}, \ldots, D_k^{(i)}, \ldots, D_k^{(n)} \right]$$

and $\{e_1, e_2, \dots, e_n\}$ is the canonical basis of $\mathbb{R}^n$, the i-th update component is

$$D_{k+1}^{(i)} = D_k^{(i)} + \frac{(\delta_k^\top e_i)^2}{\gamma_k^\top \delta_k} - \frac{(D_k^{(i)}(\gamma_k^\top e_i))^2}{\gamma_k^\top D_k \gamma_k}. \tag{3.3}$$

3.2. The BFGS type of diagonal update

By taking the diagonal of the matrix obtained by updating D_k with (1.6) formula, the i-th update component is given by

$$D_{k+1}^{(i)} = D_k^{(i)} + \left[1 + \frac{\gamma_k^\top D_k \gamma_k}{\gamma_k^\top \delta_k}\right] \frac{(\delta_k^\top e_i)^2}{\gamma_k^\top \delta_k} - 2\frac{D_k^{(i)}(\delta_k^\top e_i)(\gamma_k^\top e_i)}{\gamma_k^\top \delta_k}. \tag{3.4}$$

3.3. The inverse BFGS type of diagonal update

If we use the BFGS formula for the Hessian approximation update given by

$$B_{k+1} = B_k + \frac{\gamma_k \gamma_k^\top}{\gamma_k^\top \delta_k} - \frac{B_k \delta_k \delta_k^\top B_k}{\delta_k^\top B_k \delta_k} \tag{3.5}$$

and due to the fact that

$$D_{k+1}^{-1} = \operatorname{diag}\left[(D_{k+1}^{(1)})^{-1}, (D_{k+1}^{(2)})^{-1}, \dots, (D_{k+1}^{(i)})^{-1}, \dots, (D_{k+1}^{(n)})^{-1}\right],$$

by (3.5), we have

$$D_{k+1}^{-1} = D_k^{-1} + \frac{\gamma_k \gamma_k^\top}{\gamma_k^\top \delta_k} - \frac{D_k^{-1} \delta_k \delta_k^\top D_k^{-1}}{\delta_k^\top D_k^{-1} \delta_k},$$

finally

$$D_{k+1}^{(i)} = \left(\frac{1}{D_k^{(i)}} + \frac{(\gamma_k^\top e_i)^2}{\gamma_k^\top \delta_k} - \frac{(\delta_k^\top e_i)^2}{(D_k^{(i)})^2 (\delta_k^\top D_k^{-1} \delta_k)}\right)^{-1}. \tag{3.6}$$

Algorithm 1 (The L-BFGS method with diagonal update)
Choose $\varepsilon > 0$, $x_1 \in \mathbb{R}^n$, set $S_1 = D_1 = I$ and $k = 1$
Repeat
  Starting
  If $\|g(x_k)\| < \varepsilon$ then
    $x^* = x_k$
    stop.
  Else
    Compute direction search: $d_k = -S_k \cdot g(x_k)$
    Compute the step length t_k satisfies the Wolfe conditions (1.3)–(1.4)
    Update D_k matrix according to (3.2), (3.3), (3.4) or (3.6)
    Update S_k matrix according to (2.1)
    Set $x_{k+1} = x_k + t_k d_k$
    $k \leftarrow k + 1$
  End if
End repeat

4. Convergence Analysis

Now, we show that the L-BFGS method with (3.2), (3.4) and (3.6) updates of the diagonal matrix is globally convergent on uniformly convex problems if the line search is Wolfe type. In the case of (3.3) update formula, we can deduce this analysis if the step length t_k is determined by the exact line search

$$f(x_k + t_k d_k) = \min_{t>0} f(x_k + td_k).$$

Assumptions A

(1) The function f is twice continuously differentiable.
(2) The level set $\mathcal{L} = \{x \in \mathbb{R}^n : f(x) \leq f(x_1)\}$ is convex.
(3) There exist constants M_1, $M_2 > 0$ such that

$$M_1 \|d\|^2 \leq d^\top H(x) d \leq M_2 \|d\|^2 \tag{4.1}$$

for all $d \in \mathbb{R}^n$ and all $x \in \mathcal{L}$.

Lemma 4.1. *Let x_1 be a starting point for which f satisfies Assumptions A. Let $\{x_k\}$ be generated by Algorithm 1 using* (3.2), (3.4) *or* (3.6) *update formulas for the diagonal matrix, then $\{D_k\}$ is bounded for each $k > 0$.*

Proof. We define

$$\overline{H}_k = \int_0^1 H(x_k + \tau\delta_k)\mathrm{d}\tau,$$

then

$$\gamma_k = \overline{H}_k \delta_k. \tag{4.2}$$

Thus (4.1) and (4.2) give

$$M_1 \|\delta_k\|^2 \leq \gamma_k^\top \delta_k \leq M_2 \|\delta_k\|^2. \tag{4.3}$$

Since $\overline{H}_k$ is positive definite, then

$$\frac{\|\gamma_k\|^2}{\gamma_k^\top \delta_k} = \frac{\delta_k^\top \overline{H}_k^2 \delta_k}{\delta_k^\top \overline{H}_k \delta_k} \leq \frac{M_2 \left\|\overline{H}_k^{1/2}\delta_k\right\|^2}{\left\|\overline{H}_k^{1/2}\delta_k\right\|^2}.$$

So,

$$\frac{\|\gamma_k\|^2}{\gamma_k^\top \delta_k} \leq M_2. \tag{4.4}$$

Let J_k denote the inverse of the diagonal matrix D_k, then (3.5) can be rewritten in the next form

$$J_{k+1}^{(i)} = J_k^{(i)} + \frac{(\gamma_k^\top e_i)^2}{\gamma_k^\top \delta_k} - \frac{(J_k^{(i)}(\delta_k^\top e_i))^2}{\delta_k^\top J_k \delta_k}. \tag{4.5}$$

From (4.5) and by simple expression for the trace and the determinant of the matrix (see [**3**], [**6**])

$$\begin{aligned}\sum_{i=1}^{i=n} J_{k+1}^{(i)} &= \sum_{i=1}^{i=n} J_1^{(i)} - \sum_{k=1}^{k=m} \frac{\|J_k\delta_k\|^2}{\delta_k^\top J_k \delta_k} + \sum_{k=1}^{k=m} \frac{\|\gamma_k\|^2}{\gamma_k^\top \delta_k} \\ &\leq \sum_{i=1}^{i=n} J_1^{(i)} + \sum_{k=1}^{k=m} \frac{\|\gamma_k\|^2}{\gamma_k^\top \delta_k}.\end{aligned} \tag{4.6}$$

Let j_n denote the greatest component of J_{k+1}, from (4.4), (4.6) and the boundlessness of J_k

$$j_n \leq \sum_{i=1}^{i=n} J_{k+1}^{(i)} \leq \sum_{i=1}^{i=n} J_1^{(i)} + mM_2 \leq M_3, \tag{4.7}$$

for some positive constant M_3. From (4.5) we have

$$\prod_{i=1}^{i=n} J_{k+1}^{(i)} = \left(\prod_{i=1}^{i=n} J_1^{(i)}\right) \prod_{k=1}^{k=m} \left(\frac{\gamma_k^\top \delta_k}{\delta_k^\top \delta_k} \frac{\|\delta_k\|^2}{\delta_k^\top J_k \delta_k}\right). \tag{4.8}$$

Since

$$\frac{\|\delta_k\|^2}{\delta_k^\top J_k \delta_k} = \left(\frac{\delta_k^\top J_k \delta_k}{\|\delta_k\|^2}\right)^{-1} \geq M_3^{-1} \tag{4.9}$$

and using (4.9)

$$\prod_{i=1}^{i=n} J_{k+1}^{(i)} \geq \left(\prod_{i=1}^{i=n} J_k^{(i)}\right) \left(\frac{M_1}{M_3}\right)^m \geq M_4 \tag{4.10}$$

for some positive constant M, from (4.7), (4.10) and if j_1 denotes the smallest component of J_{k+1}, then

$$j_1 \geq \frac{M_4}{j_2 \cdot \ldots \cdot j_n} \geq \frac{M_4}{(M_3)^{n-1}} = M_5, \tag{4.11}$$

when $j_2, \ldots, j_{n-1}$ denote all other components of J_{k+1}. From (4.7) and (4.11) we conclude that for every $i = 1, 2, \ldots, n$ we have

$$M_5 \leq j_i \leq M_3,$$

which implies that the diagonal matrix J_{k+1} is bounded. □

Theorem 4.1 ([**4**]). *Under assumptions A for every start point x_1. Then if $\{D_k\}$, $k = 1, 2, \ldots$ is a bounded, positive definite matrix, the Algorithm 1 generates a sequence $\{x_k\}$ which converges to unique minimum x^* and there exists a constant r, $(0 \leq r \leq 1)$ such that*

$$f(x_k) - f(x^*) \leq r^k [f(x_1) - f(x^*)], \tag{4.12}$$

i.e. the sequence $\{x_k\}$ converges to x^* at a r-linear rate.

5. Experiments with selected problems

We have focused on the five problems listed in Table 1, these problems were proposed by J. J. Moré, B. S. Garbow and K. E. Hillstrome in [**5**]. We used FORTRAN 90 code package documented and available in [**9**]. For the line searches we used $c_1 = 0.3$, $c_2 = 0.7$ for Wolfe line search and $c_1 = 0.3$ for Armijo line search (the step length satisfies only (1.3) condition [**1**], [**6**]). The optimization iteration was terminated when

$$\|g(x_k)\| \leq \varepsilon, \qquad (\varepsilon \approx 10^{-8})$$

using $m = 5$ and the number of variable n on all these test problems is between 500 and 10,000.

Table 1. Table of test problems.

Problems	Problem's name
P.I	Extended Dixon Function
P.II	Extended Oren Function
P.III	Extended Powell Singular Function
P.IV	Extended Rosenbrock Function
P.V	Extended Wood Function

In the following tables we compare the performance of the L-BFGS method with Oren-Spedicato scaling and others updates where of the diagonal matrix proposed in this work (DFP, BFGS and Inverse BFGS) n denotes the number of variables and the results are reported in the form: number of iterations/total time.

Table 2. L-BFGS with Armijo line search.

Problems	n	Oren-Sped.	DFP	BFGS	Inverse BFGS
P.I	500	360/1.53	398/1.76	382/1.50	355/1.48
	1000	575/2.86	558/2.91	566/2.66	533/2.01
	5000	1437/17.41	1328/19.41	1443/17.36	1192/14.82
	10000	1909/42.26	1720/42.33	1922/40.02	1754/32.62
P.II	500	47/1.68	342/8.68	–	49/1.64
	1000	51/4.89	496/24.84	–	57/4.85
	5000	229/208.15	–	–	611/208.48
	10000	301/970.43	–	–	298/763.52
P.III	500	170/0.25	341/0.36	341/0.29	246/0.28
	1000	365/2.92	353/5.76	–	357/2.20
	5000	536/6.43	3880/9.22	–	356/4.72
	10000	477/9.89	335/20.43	–	539/10.85

Problems	n	Oren-Sped.	DFP	BFGS	Inverse BFGS
P.IV	500	36/0.16	37/0.60	35/0.11	38/0.16
	1000	36/1.17	36/0.50	33/0.16	36/0.14
	5000	37/0.44	38/1.98	33/0.40	38/0.45
	10000	37/0.77	37/3.24	36/0.78	36/0.70
P.V	500	108/0.60	48/0.82	58/0.27	64/0.27
	1000	69/0.50	90/0.61	66/0.55	50/0.33
	5000	72/1.98	79/3.13	57/1.43	102/1.59
	10000	72/3.24	67/3.02	39/0.987	67/1.92

Table 3. L-BFGS with Wolfe line search.

Problems	n	Oren-Sped.	DFP	BFGS	Inverse BFGS
P.I	500	372/1.57	344/1.37	359/1.55	340/1.19
	1000	491/2.7	496/3.79	503/3.10	544/3.05
	5000	1480/15.18	1353/23.67	1480/15.13	1194/12.90
	10000	2153/58.61	1911/56.08	2103/52.57	1935/41.79
P.II	500	39/5.22	–	–	39/5.01
	1000	88/23.45	–	–	95/23.50
	5000	178/839.59	–	–	422/839.54
	10000	329/1079.55	–	–	402/1022.65
P.III	500	204/0.99	103/0.93	131/0.77	138/0.75
	1000	298/5.83	301/3.84	254/2.50	282/2.47
	5000	561/10.49	543/9.99	–	484/9.40
	10000	519/11.35	713/20.54	–	461/10.03
P.IV	500	35/0.17	29/0.16	36/0.16	35/0.14
	1000	36/0.22	35/0.22	34/0.21	36/0.21
	5000	36/0.81	35/1.27	53/0.83	34/0.76
	10000	36/1.65	36/2.04	35/1.62	36/1.61
P.V	500	80/0.33	64/0.32	54/0.27	90/0.33
	1000	82/0.82	70/0.83	54/0.70	95/0.77
	5000	82/1.59	67/1.97	53/1.41	95/1.56
	10000	59/2.64	37/3.35	46/2.12	52/2.01

6. Conclusions

Our numerical study indicates that updating the diagonal matrix of the L-BFGS with inverse BFGS performs better than the Oren-Spedicato scaling (3.2), and the BFGS update (3.4) performes well sometimes, but is inefficient in some cases. The numerical and theoretical results lead to reject the DFP formula (3.3). It should be noted that in generally the Wolfe line search gives better results than the Armijo line search type. Finally, it is not clear that the same conclusion would be valid for the case of the constrained optimization problems, but this is certainly an issue that deserves further research.

References

1. Denis J. and Schnabel R. B., *Numérical method for unconstrained optimization and nonlinear equation.* Prentice Hall, 1983.

2. Gill P. E. and Murray W., *Conjugate gradient method for large-scale nonlinear optimization,* Technical report SOL 79–15, Dept. of Operation Res. Stanford University, 1979.

3. Kolda T. G., O'leary D. P. and Nazareth L., *BFGS with update skipping and varying memory*, SIAM J. Optim. **8(4)** (1998), 1060–1083.

4. Liu D. C. and Nocedal J., *On the limited memory BFGS method for large scale optimization*, Math. Prog. **45** (1989), 503–528.

5. Moré J. J., Garbow B. S. and Hillstrome K. E., *Testing unconstrained optimization software*, ACM Trans. Math. Software, 7 (1981), 17–41.

6. Nocedal J., *Theory of algorithms for unconstrained optimization*, Acta Numerica, **1** (1999), 199–242.

7. Oren S. S. and Spedicato E., *Optimal condition of self-scaling variable metric algorithms*, Math. Prog., **10** (1976), 70–90.

8. Powell M. J. D., *Some properties of the variable metric algorithm for minimization without exact line searches*, in Nonlinear Programming, SIAM-AMS Proceeding **IX**, R.W. Cottle, and C.E. Lemke, eds., SIAM, 35–72, 1976.

9. Sahari M. L., *FORTRAN 90 software for nonlinear unconstrained optimization, home page*, `http://sahari.site.voila.fr/OptimizationCodeF90.html`

M. L. Sahari, Département de Mathématiques, Université Badji-Mokhtar, Annaba, Algérie,
e-mail: `mlsahari@yahoo.fr`

R. Khaldi, Département de Mathématiques, Université Badji-Mokhtar, Annaba, Algérie,
e-mail: `rkhadi@yahoo.fr`