A Comparison between LBFGS and a free self-scaling VM algorithms for unconstrained optimization

Omar B. Mohammed
Dept. of Mathematics, College of Science, University of Koya, Erbil, Iraq

(**Received:** 28 / 9 / 2009 ---- **Accepted:** 23 / 5 / 2010)

Abstract

In this paper, a comparison between the limited memory BFGS algorithm, LBFGS developed by Nocedal (1980) and free self-scaling VM algorithm called (MBFGS) algorithm has been investigated.
The free self-scaling VM algorithms is the best due to its low storage requirement and also able to solve large-scale test problems with 10^6 variables successfully while the other method fail to converge in this accuracy.

1. Introduction

Quasi-Newton (QN) methods for unconstrained optimization are a class of numerical techniques for solving the following problem

$$\min_x f(x) \quad (1)$$

where $f(x)$ is a nonlinear real-valued function and x is an n-dimensional real vector. At the kth iteration, an approximation point x_k and an nxn matrix H_k are available. The methods proceed by generating a sequence of approximation points due to the equation

$$x_{k+1} = x_k + \lambda_k d_k \quad (2)$$

Where λ_k is calculated to satisfy certain line search conditions and d_k is a descent direction.

One important feature of QN methods is the choice of the matrix H_k. The methods require H_k to be positive definite and satisfy the QN equation

$$H_{k+1} y_k = \alpha_k s_k , \alpha_k > 0 \quad (3)$$

Where, $s_k = x_{k+1} - x_k$ and $y_k = g_{k+1} - g_k$ with g the gradient of f. One of the best known QN methods is the BFGS method that was proposed independently by Broyden [1], Fletcher [2], Goldfarb [3] and Shanno [4]. The BFGS update is defined by the equation

$$H_{k+1} = H_k + \frac{1}{s_k^T y_k}\left[\left(\alpha_k + \frac{y_k^T H_k y_k}{s_k^T y_k}\right) s_k s_k^T - s_k y_k^T H_k - H_k y_k s_k^T\right] \quad (4)$$

With $\alpha_k = 1$.

The BFGS update has been used successfully in many production codes for solving unconstrained optimization problems. In practice, we observe from numerical results of many other papers (see Luksan [5] for instance) that the BFGS out-performed many QN updates in solving practical problems.

Our interest here is the limited memory extension to the QN methods, which will suit for the solution of large-scale optimization problems.

Limited memory QN methods has been considered by Nocedal [6], where it is called the SQN method. The user specifies the number m of QN (BFGS, for instance) corrections that are to be kept, and provides a sparse symmetric and positive definite matrix H_0, which approximates the inverse Hessian of f. During the first m iterations the method is identical to the QN method. For $k > m$, H_k is obtained by applying m QN updates to H_0 using information from the m previous iterations. The limited memory BFGS method (LBFGS) by Nocedal uses the inverse BFGS formula in the form

$$H_{k+1} = V_k^T H_k V_k + \rho_k s_k s_k^T \quad (5)$$

where

$$\rho_k = 1/ y_k^T s_k , \quad V_k = I - \rho_k y_k s_k^T \quad (6)$$

(see Dennis and Schnabel [7].)

2. Updating BFGS algorithm: [8]

To improve the performance of the BFGS updates, Biggs [9] first suggested a self-adjustable value for the parameter α_k. Based upon non-quadratic models, he derived the parameter α_k as

$$\alpha_k = 1/t_k \quad (7)$$

where

$$t_k = \frac{6}{s_k^T y_k}\left(f(x_k - f(x_{k+1} + s_k^T g_{k+1}\right) - 2 \quad (8)$$

Hence, a modified BFGS update can be defined by

$$H_{k+1} = H_k + \frac{1}{s_k^T y_k}\left(\alpha_k + \frac{y_k^T H_k y_k}{s_k^T y_k}\right) s_k s_k^T - s_k y_k^T H_k - H_k y_k s_k^T \quad (9)$$

and a new modified BFGS update can be written as:

$$\alpha_k = y^T s / y^T y \quad (10)$$ and

$$H_{k+1} = H_k + \frac{1}{s_k^T y_k} * \left(\left(\frac{1}{\alpha_k}\right)^2 + \frac{y_k^T H_k y_k}{s_k^T y_k}\right) s_k s_k^T - s_k y_k^T H_k - H_k y_k s_k^T \quad (11)$$

A more useful form of (9) when apply to large-scale problems can be written as follows:

$$H_{k+1} = v_k^T H_k v_k + \alpha_k \rho_k s_k s_k^T \quad (12)$$

Therefore, given any initial approximate inverse Hessian H_0, a recursive formula for (12) at any iteration k can be expressed as follows:

$$\begin{aligned} H_{k+1} &= (v_k^T....v_0^T)H_0(v_0....v_k) \\ &+ \alpha_0\rho_0(v_k^T....v_1^T)s_0s_0^T(v_1....v_k) \\ &+ \alpha_1\rho_1(v_k^T....v_2^T)s_1s_1^T(v_2....v_k) \\ &. \\ &. \\ &. \\ &+ \alpha_k\rho_k s_k s_k^T \end{aligned} \quad (13)$$

3.Convergence of the BFGS method:

a. The Global Convergence of the BFGS method:

We study the global convergence of BFGS, with a practical line search, when applied to a smooth convex function from an arbitrary starting point x_0 and from any initial Hessian approximation B_0 that is symmetric and positive definite.

Theorem (1): Let B_0 be any symmetric positive definite initial matrix, and let x_0 be a starting point. Then the sequence $\{x_k\}$ converges to the minimizer x^* of f.

Proof: Let we define

$$m_k = \frac{y_k^T s_k}{s_k^T s_k}. \quad (14)$$

$$M_k = \frac{y_k^T y_k}{y_k^T s_k}. \quad (15)$$

such that

$$\frac{y_k^T s_k}{s_k^T s_k} = \frac{s_k^T G_k^- s_k}{s_k^T s_k} \geq m \quad (16)$$

where G_k^- is the average Hessian defined as:

$$G_k = \left[\int_0^1 \nabla^2 f(x_k + \tau\alpha_k\rho_k)d\tau\right]$$

and we know that G_k^- is positive definite, so its square root is well defined. Therefore there exists a square root $G_k^{-(1/2)}$ satisfying:

$G_k^- = G_k^{-(1/2)}\, G_k^{-(1/2)}$. Therefore, by defining $z_k = G_k^{-(1/2)} s_k$, and using the property:

$$y_k = G^-\alpha_k\rho_k = G^- s_k \quad (17)$$

we have:

$$\frac{y_k^T s_k}{y_k^T y_k} = \frac{\left(G_k^{-(1/2)}s_k\right)^T G_k^{-(1/2)}s_k}{\left(G_k^{-(1/2)}s_k\right)^T G_k^- G_k^{-(1/2)}s_k} = \frac{z_k^T z_k}{z_k^T G_k^- z_k} \quad (18)$$

then

$$\frac{y_k^T y_k}{y_k^T s_k} = \frac{z_k^T G_k^- z_k}{z_k^T z_k} \leq M \quad (19)$$

by computing the trace of the BFGS approximation

$$B_{k+1} = B_k - \frac{B_k s_k s_k^T B_k}{s_k^T B_k s_k} + \frac{y_k y_k^T}{y_k^T s_k}. \quad (20)$$

we obtain that

$$trace(B_{k+1}) = trace(B_k) - \frac{\|B_k s_k\|^2}{s_k^T B_k s_k} + \frac{\|y_k\|^2}{y_k^T s_k} \quad (21)$$

and let we define:

$$\cos\theta_k = \frac{s_k^T B_k s_k}{\|s_k\|\|B_k s_k\|},$$

$$q_k = \frac{s_k^T B_k s_k}{s_k^T s_k}. \quad (22)$$

so that θ_k is the angle between s_k and $B_k s_k$ .we then obtain that:

$$\frac{\|B_k s_k\|^2}{s_k^T B_k s_k} = \frac{\|B_k s_k\|^2\|s_k\|^2}{\left(s_k^T B_k s_k\right)^2} * \frac{s_k^T B_k s_k}{\|s_k\|^2} = \frac{q_k}{\cos^2\theta_k} \quad (23)$$

from (14) and (15) we have:

$$\det(B_{k+1}) = \det(B_k).\frac{y_k^T s_k}{s_k^T s_k} * \frac{s_k^T s_k}{s_k^T B_k s_k} = \det(B_k).\frac{m_k}{q_k} \quad (24)$$

we now combine the trace and determinant by introducing the following function of a positive definite matrix B:

$$\varphi(B) = trace(B) - \ln(\det(B)) \quad (25)$$

where ln (.) denotes the natural logarithm. It is not difficult to show that $\varphi(B) > 0$.

by using (14), (15), (21)-(25), we have that:

$$\varphi(B_{k+1}) = \varphi(B_k) + M_k - \frac{q_k}{\cos^2\theta_k} - $$
$$-\ln(\det(B_k)) - \ln m_k + \ln q_k$$
$$= \varphi(B_k) + (M_k - \ln m_k - 1) \quad (26)$$
$$+\left[1 - \frac{q_k}{\cos^2\theta_k} + \ln\frac{q_k}{\cos^2\theta_k}\right]$$
$$+\ln\cos^2\theta_k$$

Now, since the function $h(t) = 1 - t + \ln t \le 0$ is non-positive for all $t > 0$, the term inside the square brackets is non-positive, and thus

$$0 < \varphi(B_{k+1}) \le \varphi(B_1) + ck + + \sum_{j=1}^{k} \ln\cos^2\theta_k \quad (27)$$

where we can assume constant $c = M - \ln m - 1$ to be positive, without loss of generality.

Note from the form $s_k = -\alpha_k B_k^{-1}\nabla f_k$ of the quasi-Newton iteration that $\cos\theta_k$ defined by (22) is the angle between the steepest descent direction and the search direction.

we know that the sequence $\|\nabla f_k\|$ generated by the line search algorithm is bounded away from zero only if $\cos\theta_j \to 0$.

Assume that $\cos\theta_j \to 0$. Then there exist $k_1 > 0$ such that for all $j > k$,

we have

$$\ln\cos^2\theta_j < -2c,$$

where c is the constant defined above. Using the inequality in (27) we find the following relation to be true for all $k > k_1$:

$$0 < \varphi(B_1) + ck + \sum_{j=1}^{k_1} \ln\cos^2\theta_j + \sum_{j=k_1+1}^{k}(-2c) \quad (28)$$
$$= \varphi(\mathrm{B}_1) + \sum_{j=1}^{k_1} \ln\cos^2\theta_j + 2ck_1 - ck$$

However, the right hand side is negative for large k, giving a contradiction.

therefore, there exists a subsequence of indices $\{j_k\}$ such that $\{\cos\theta_{jk} \ge \delta > 0\}$, and implies that $\liminf\|\nabla f_k\| \to 0$.

since the problem is strongly convex, this enough to prove that $x_k \to x^*$.

b. Superlinear convergence of BFGS method:

Theorem (2): suppose that f is twice continuously differentiable, and that the iterates generated by the BFGS algorithm converge to a minimizer x^*. Suppose that

$\sum_{k=1}^{\infty}\|x_k - x^*\| < \infty$ hold. Then x_k converges to x^* at a superlinear rate.

Proof: from the following relation:

$$\frac{\left\|\tilde{y}_k - \tilde{s}_k\right\|}{\left\|\tilde{s}_k\right\|} \le \tilde{c}\,\varepsilon_k \quad (29)$$

we have from the triangle inequality,

$$\left\|\tilde{y}_k\right\| - \left\|\tilde{s}_k\right\| \le \bar{c}\,\varepsilon\left\|\tilde{s}_k\right\|,$$
$$\left\|\tilde{s}_k\right\| - \left\|\tilde{y}_k\right\| \le \bar{c}\,\varepsilon_k\left\|\tilde{s}_k\right\|.$$

so that

$$(1 - \bar{c}\,\varepsilon_k)\left\|\tilde{s}_k\right\| \le \left\|\tilde{y}_k\right\| \le (1 + \bar{c}\,\varepsilon_k)\left\|\tilde{s}_k\right\| \quad (30)$$

by squaring (29) and using (30), we obtain

$$(1 - \bar{c}\,\varepsilon_k)^2\left\|\tilde{s}_k\right\|^2 - 2\tilde{y}_k^T\tilde{s}_k +$$
$$\left\|\tilde{s}_k\right\|^2 \le \left\|\tilde{y}_k\right\|^2 - 2\tilde{y}_k^T\tilde{s}_k +$$
$$\left\|\tilde{s}_k\right\|^2 \le \bar{c}^2\ \varepsilon_k^2\left\|\tilde{s}_k\right\|^2$$

and therefore

$$2\tilde{y}_k^T\tilde{s}_k \ge (1 - 2\bar{c}\,\varepsilon_k + \bar{c}^2\ \varepsilon_k^2 + 1 - \bar{c}^2\ \varepsilon_k^2)\left\|\tilde{s}_k\right\|^2$$
$$= 2(1 - \bar{c}\,\varepsilon_k)\left\|\tilde{s}_k\right\|^2$$

it follows from the definition of $\tilde{m}_k$ that

$$\tilde{m}_k = \frac{\tilde{y}_k^T\tilde{s}_k}{\left\|\tilde{s}_k\right\|^2} \ge 1 - \bar{c}\,\varepsilon_k \quad (31)$$

by combining (30) and (31), we obtain also that:

$$\tilde{M}_k = \frac{\left\|\tilde{y}_k\right\|^2}{\tilde{y}_k\tilde{s}_k} \le \frac{1 + \bar{c}\varepsilon_k}{1 - \bar{c}\varepsilon_k}, \quad (32)$$

since $x_k \to x^*$, we have that $\varepsilon_k \to 0$, and thus by (32) there exists a positive constant $c > \bar{c}$ such that the following inequalities hold for all large k:

$$\tilde{M}_k \le 1 + \frac{2\bar{c}}{1-\bar{c}\varepsilon_k}\varepsilon_k \le 1 + c\varepsilon_k \qquad (33)$$

we again make use of the nonpositiveness of the function $h(t) = 1 - t + \ln t$.

therefore we have

$$\frac{-x}{1-x} - \ln(1-x) = h\left(\frac{1}{1-x}\right) \le 0$$

now for k large enough we can assume that $\bar{c}\,\varepsilon_k < 1/2$, and therefore

$$\ln\left(1 - \bar{c}\,\varepsilon_k\right) \ge \frac{-\bar{c}\,\varepsilon_k}{1-\bar{c}\,\varepsilon_k} \ge -2\bar{c}\,\varepsilon_k. \qquad (34)$$

this relation and (31) imply that for sufficiently large k, we have

$$\ln \tilde{m}_k \ge \ln\left(1 - \bar{c}\,\varepsilon_k\right) \ge -2\bar{c}\,\varepsilon_k > -2c\varepsilon_k.$$

we can get from (26), (33) and (34)

$$0 < \varphi(\tilde{B}_{k+1}) \le \varphi(\tilde{B}_k) + 3c\varepsilon_k + \ln\cos^2\tilde{\theta}_k + \\ + \left[1 - \frac{\tilde{q}_k}{\cos^2\tilde{\theta}_k} + \ln\frac{\tilde{q}_k}{\cos^2\tilde{\theta}_k}\right]$$

(35)

by using (35) and using $\sum_{k=1}^{\infty}\|x_k - x^*\| < \infty$ we have:

$$\sum_{j=0}^{\infty}\left(\ln\frac{1}{\cos^2\tilde{\theta}_j} - \left[1 - \frac{\tilde{q}_j}{\cos^2\tilde{\theta}_j} + \ln\frac{\tilde{q}_j}{\cos^2\tilde{\theta}_j}\right]\right).$$

$$\le \varphi(\tilde{B}_0) + 3c\sum_{j=0}^{\infty}\varepsilon_j < +\infty$$

since the term in the square brackets is non positive, and since $1/\cos^2\tilde{\theta}_j \ge 0$ for all j, we obtain:

$$\lim_{j\to\infty}\ln\frac{1}{\cos^2\tilde{\theta}_j} = 0,$$

$$\lim_{j\to\infty}\left(1 - \frac{\tilde{q}_j}{\cos^2\tilde{\theta}_j} + \ln\frac{\tilde{q}_j}{\cos^2\tilde{\theta}_j}\right) = 0$$

which imply that

$$\lim_{j\to\infty}\cos\tilde{\theta}_j = 1, \quad \lim_{j\to\infty}\cos\tilde{q}_j = 1 \qquad (36)$$

From (23), we have

$$\frac{\left\|G_*^{-1/2}(B_k - G_*)s_k\right\|^2}{\left\|G_*^{1/2}s_k\right\|^2} = \frac{\left\|(\tilde{B}_k - I)\tilde{s}_k\right\|^2}{\left\|\tilde{s}_k\right\|^2}$$

$$= \frac{\left\|\tilde{B}_k\tilde{s}_k\right\|^2 - 2\tilde{s}_k^T\tilde{B}_k\tilde{s}_k + \tilde{s}_k^T\tilde{s}_k}{\tilde{s}_k^T\tilde{s}_k}$$

$$= \frac{\tilde{q}_k^2}{\cos\tilde{\theta}_k^2} - 2\tilde{q}_k + 1$$

from (36) the right hand side converges to 0, we get

$$\lim_{k\to\infty}\frac{\|(B_k - G_*)s_k\|}{\|s_k\|} = 0$$

And hence the rate of convergence is superlinear.

4. Modified BFGS method:

Based upon the recursive formula (13) and limited memory updating procedures developed by Nocedal [6], we can now state a limited memory modified BFGS algorithm with inexact line searches as follows [10]:

Algorithm 1: New BFGS method

Step 1. Choose x_0, $0 < \beta' < 1/2$, $\beta' < \beta < 1$, and initial matrix $H_0 = I$, Set $k = 0$.

Step 2. Compute

$d_k = -H_k g_k$, and $x_{k+1} = x_k + \lambda_k d_k$

where λ_k satisfies

$$f(x_k + \lambda_k d_k) \le f(x_k) + \beta'\lambda_k g_k^T d_k$$

$$g(x_k + \lambda_k d_k)^T d_k \ge \beta g_k^T d_k$$

(the steplength $\lambda_k = 1$ is tried first).

Step 3. Let $\hat{m} = \min\{m, k-1\}$ Update H_0 for $\hat{m}+1$ times by using the pair $\{y_j, s_j\}_{j=k-\hat{m}}^{k}$ i.e. let

$$H_{k+1} = (v_k^T \dots v_{k-\hat{m}}^T)H_0(v_{k-\hat{m}} \dots v_k)$$
$$+ \alpha_{k-\hat{m}}\rho_{k-\hat{m}}(v_k^T \dots v_{k-\hat{m}+1}^T)s_{k-\hat{m}}s_{k-\hat{m}}^T(v_1 \dots v_k)$$
$$+ \alpha_{k-\hat{m}+1}\rho_{k-\hat{m}+1}(v_k^T \dots v_{k-\hat{m}+2}^T)s_{k-\hat{m}+1}s_{k-\hat{m}+1}^T(v_{k-\hat{m}+2} \dots v_k)$$
$$\vdots$$
$$+ \alpha_k\rho_k s_k s_k^T \qquad (37)$$

with α_k calculated by (7).

In order to obtain the new steps and a new algorithms for the modified BFGS we suggest the following:
To calculate the approximation inverse Hessian, H using (37), we need additional $\hat{m}$ storage for α. We try to avoid this by introducing new updating formula in Step 3 of Algorithm 1 as follows:

$$H_{k+1} = (v_k^T \dots v_{k-\hat{m}}^T) H_0 (v_{k-\hat{m}} \dots v_k)$$
$$+ \rho_{k-\hat{m}} (v_k^T \dots v_{k-\hat{m}+1}^T) s_{k-\hat{m}} s_{k-\hat{m}}^T (v_{k-\hat{m}+1} \dots v_k)$$
$$+ \rho_{k-\hat{m}+1} (v_k^T \dots v_{k-\hat{m}+2}^T) s_{k-\hat{m}+1} s_{k-\hat{m}+1}^T (v_{k-\hat{m}+2} \dots v_k)$$
$$\vdots$$
$$+ \alpha_k \rho_k s_k s_k^T \qquad (38)$$

By doing so, we only need to calculate α_k instead of $\alpha_{k-\hat{m}}, \alpha_{k-\hat{m}+1}, \dots, \alpha_k$. Therefore formula (38) is preferred.
Step 4. Set $k = k+1$, and go to Step 2.

Algorithm 2: MBFGS method (the new modified BFGS algorithm):

step 1: Choose x_0 as initial point, and initial matrix $H_0 = I$

step 2: Let $\varepsilon_0 > 0$, $k = 0$, repeat.

step 3: Compute $d_k = -H_k g_k$, and $x_{k+1} = x_k + \lambda_k d_k$,

where λ_k satisfies the following conditions:

$$f(x_k + \lambda_k d_k) \le f(x_k) + \beta' \lambda_k g_k^T d_k$$
$$g(x_k + \lambda_k d_k)^T d_k \ge \beta g_k^T d_k$$

step 4: Compute $s_k = x_{k+1} - x_k$, $y_k = g_{k+1} - g_k$.

step 5: Compute H_k from

$$H_{k+1} = H_k + \frac{1}{s_k^T y_k} * \left(\left(\frac{1}{\alpha_k} \right)^2 + \frac{y_k^T H_k y_k}{s_k^T y_k} \right) s_k s_k^T - s_k y_k^T H_k - H_k y_k s_k^T$$

s.t $\alpha_k = \dfrac{y_k^T y_k}{y_k^T s_k}$

and

$$H_{k+1} = (V_k^T \dots V_{k-3}^T) H_0 (V_{k-3} \dots V_k)$$
$$+ \rho_{k-3} (V_k^T \dots V_{k-2}^T) s_{k-3} s_{k-3}^T (V_{k-2} \dots V_k)$$
$$+ \rho_{k-2} (V_k^T \dots V_{k-1}^T) s_{k-2} s_{k-2}^T (V_{k-1} \dots V_k)$$
$$+ \alpha_k \rho_k s_k s_k^T$$

step 6: If $\|g_{k+1}\| < \varepsilon$ then stop, otherwise, put $k = k+1$ and Goto step (3).

As can be seen from Tables 1-10, the problems especially problems with large number of variables, the number of iterations required by the new MBFGS algorithm is less than the corresponding numbers required by LBFGS. This indicates that the new method is a better choice for solving large-scale optimization problems.

4.1: Another family of VM updates was proposed by bigges (1971) for use in function minimization algorithms. Biggs family takes non-quadratics of the objective function into account in order to obtain a better approximation to B than that given by either DFP or BFGS updates.

However, Bigges (1973) observed that a more accurate estimate of this curvature can be obtained using for independent pieces of information which are available along v_k, namely the function values and directional derivatives at the two successive points. The following cubic model of f along v_k was contructed by bigges for which:

$$\rho_k = \frac{v_k^T y_k}{4 v_k^T g_{k+1} + 2 v_k^T g_k - 6(f_{k+1} - f_k)}. \quad (39)$$

and the updating formula H_k has the form:

$$H_{k+1}^{Bigges} = H_k - \frac{H_k y_k y_k^T}{y_k^T H_k y_k} + \rho_k \frac{v_k v_k^T}{v_k^T y_k} + w_k w_k^T. \quad (40)$$

where:

$$w_k = (y_k^T H_k y_k)^{1/2} \left\{ \frac{v_k}{v_k^T y_k} - \frac{H_k y_k}{y_k^T H_k y_k} \right\} \qquad (41)$$

Table (4.1) A: (Comparison between standard LBFGS method and MBFGS using Rosen test function (2≤n≤100000)

Rosen	LBFGS $\rho = 1$		MBFGS	
Function	NOI	Function Value	NOI	Function Value
n=2	39	7.1D-17	29	7.5D-17
n=4	37	1.5D-13	28	8.2D-16
n=8	38	1.3D-12	27	5.5D-14
n=10	37	2.0D-14	27	4.8D-14
n=40	35	1.7D-16	29	1.2D-13
n=100	35	2.1D-13	29	1.3D-13
n=1000	37	3.4D-12	23	5.5D-15
n=5000	35	1.1D-9	32	2.8D-14
n=10000	37	6.8D-13	29	2.5D-12
n=100000	36	6.9D-10	31	2.3D-10
Total	366		284	

Table (4.1) B: Percentage performance of the new modification against the standard ethod using Rosen function

Tools	BFGS $\rho = 1$	MBFGS
NOI	100%	77.6%

Table (4.2) A: (Comparison between standard LBFGS method and MBFGS using Shallow test function (2≤n≤100000)

Shallow	LBFGS $\rho = 1$		MBFGS	
Function	NOI	Function Value	NOI	Function Value
n=2	12	2.0D-11	11	1.6D-13
n=4	13	2.6D-14	12	1.3D-11
n=8	13	1.4D-13	14	1.1D-13
n=10	12	1.7D-11	13	9.4D-16
n=40	14	5.0D-13	11	2.1D-9
n=100	13	1.0D-11	11	5.3D-16
n=1000	14	1.2D-11	11	5.4D-8
n=5000	13	5.7D-8	11	2.0D-15
n=10000	13	7.5D-10	14	1.9D-8
n=100000	12	2.0D-8	10	4.2D-8
Total	129		118	

Table (4.2) B: Percentage performance of the new modification against the standard method using Shallow function

Tools	BFGS $\rho = 1$	MBFGS
NOI	100%	91.47%

Table (4.3) A: (Comparison between standard LBFGS method and MBFGS using Cubic test function (2≤n≤100000)

Cubic	LBFGS $\rho = 1$		MBFGS	
Function	NOI	Function Value	NOI	Function Value
n=2	38	1.6D-19	17	2.6D-14
n=4	37	2.7D-13	18	1.6D-18
n=8	36	2.8D-16	22	1.0D-9
n=10	37	4.7D-18	22	4.4D-18
n=40	36	2.1D-14	23	3.2D-14
n=100	39	6.8D-13	20	4.4D-14
n=1000	38	1.1D-11	19	1.2D-14
n=5000	38	3.9D-10	23	2.2D-12
n=10000	40	5.1D-13	22	3.1D-13
n=100000	40	2.7D-11	21	2.3D-10
Total	379		207	

Table (4.3) B: Percentage performance of the new modification against the standard method using Cubic function

Tools	BFGS $\rho = 1$	MBFGS
NOI	100%	54.62%

Table (4.4) A: (Comparison between standard LBFGS method and MBFGS using Wood test function (2≤n≤100000)

Wood	LBFGS $\rho = 1$		MBFGS	
Function	NOI	Function Value	NOI	Function Value
n=2	17	9.0D+3	13	9.0D+3
n=4	94	2.0D-17	91	2.7D-15
n=8	92	8.9D-13	30	2.2D-15
n=10	135	9.0D+3	116	9.0D+3
n=40	93	1.1D-12	27	1.0D-14
n=100	89	1.2D-11	23	5.5D-16
n=1000	95	3.8D-14	27	2.6D-13
n=5000	95	1.4D-10	27	1.0D-10
n=10000	91	1.3D-10	25	3.2D-12
n=100000	90	5.8D-9	25	1.3D-14
Total	891		404	

Table (4.4) B: Percentage performance of the new modification against the standard method using Wood function

Tools	LBFGS $\rho = 1$	MBFGS
NOI	100%	45.34%

Table (4.5) A: (Comparison between standard LBFGS method and MBFGS using Edgar test function (2≤n≤100000)

Edgar	LBFGS $\rho = 1$		MBFGS	
Function	NOI	Function Value	NOI	Function Value
n=2	7	2.3D-11	7	3.0D-11
n=4	7	1.9D-10	6	1.9D-11
n=8	8	1.0D-12	6	6.2D-12
n=10	8	2.4D-13	6	1.2D-12
n=40	8	2.6D-10	7	6.3D-11
n=100	8	2.7D-9	9	9.1D-13
n=1000	8	6.5D-9	7	1.5D-9
n=5000	8	2.8D-8	7	4.4D-8
n=10000	8	3.7D-7	8	1.3D-10
n=100000	8	6.0D-7	7	1.5D-6
Total	78		70	

Table (4.5) B: Percentage performance of the new modification against the standard method using Edgar function

Tools	LBFGS $\rho = 1$	MBFGS
NOI	100%	89.74%

Table (4.6) A: (Comparison between standard LBFGS method and MBFGS using Powell test function (2≤n≤100000)

Powell	LBFGS $\rho = 1$		MBFGS	
Function	NOI	Function Value	NOI	Function Value
n=2	---	---	---	---
n=4	---	---	---	---
n=8	51	1.6D-11	39	1.4D-11
n=10	49	1.0D-10	39	1.4D-11
n=40	36	4.8D-10	39	5.3D-10
n=100	56	3.4D-11	44	4.8D-12
n=1000	54	2.8D-10	49	1.1D-11
n=5000	46	1.9D-11	44	8.8D-12
n=10000	60	1.4D-10	47	4.1D-14
n=100000	67	4.1D-10	56	2.9D-12
Total	419		357	

Table (4.6) B: Percentage performance of the new modification against the standard method using Powell function

Tools	LBFGS $\rho = 1$	MBFGS
NOI	100%	85.2%

Table (4.7) A: (Comparison between standard LBFGS method and MBFGS using Pen2 test function (2≤n≤100000)

Pen2	LBFGS $\rho = 1$		MBFGS	
Function	NOI	Function Value	NOI	Function Value
n=2	3	5.0D-6	6	5.0D-6
n=4	3	1.0D-5	2	1.0D-5
n=8	7	2.0D-5	5	2.0D-5
n=10	7	2.5D-4	6	2.5D-4
n=40	8	1.2D-4	5	1.2D-4
n=100	8	2.5D-4	6	2.5D-4
n=1000	8	2.5D-3	6	2.5D-3
n=5000	8	1.2D-2	5	1.2D-2
n=10000	8	2.5D-2	7	2.5D-2
n=100000	8	2.5D-1	6	2.5D-1
Total	68		54	

Table (4.7) B: Percentage performance of the new modification against the standard method using Pen2 function

Tools	LBFGS $\rho = 1$	MBFGS
NOI	100%	79.41%

Table (4.8) A: (Comparison between standard LBFGS method and MBFGS using Dixon test function (2≤n≤100000)

Dixon	LBFGS $\rho = 1$		MBFGS	
Function	NOI	Function Value	NOI	Function Value
n=2	10	6.1D-13	7	5.7D-16
n=4	17	2.1D-11	12	2.1D-14
n=8	40	6.6D-11	22	2.5D-11
n=10	53	4.7D-11	26	1.5D-11
n=40	212	5.0D-1	31	3.4D-10
n=100	185	5.0D-1	32	2.2D-10
n=1000	205	5.0D-1	27	7.5D-9
n=5000	191	5.0D-1	50	6.3D-8
n=10000	171	5.0D-1	74	4.8D-8
n=100000	178	5.0D-1	74	9.5D-7
Total	1262		355	

Table (4.8) B: Percentage performance of the new modification against the standard method using Dixon function

Tools	LBFGS $\rho = 1$	MBFGS
NOI	100%	28.13%

Table (4.9) A: (Comparison between standard LBFGS method and MBFGS using Fox test function (2≤n≤100000)

Fox	LBFGS $\rho = 1$		MBFGS	
Function	NOI	Functio n Value	NOI	Function Value
n=2	12	-5.1D-1	8	-5.1D-1
n=4	10	-1.0D0	8	-1.0D0
n=8	12	-2.0D0	8	-2.0D0
n=10	12	-2.0D0	8	-2.0D0
n=40	13	-1.0D+1	11	-1.0D+1
n=100	12	-2.5D+1	8	-2.5D+1
n=1000	13	-2.5D+2	11	-2.5D+2
n=5000	12	-1.2D+3	10	-1.2D+3
n=10000	13	-2.5D+3	11	-2.5D+3
n=100000	12	-2.5D+4	10	-2.5D+4
Total	121		93	

Table (4.9) B: Percentage performance of the new modification against the standard method using Fox function

Tools	LBFGS $\rho = 1$	MBFGS
NOI	100%	76.86%

Table (4.10) A: (Comparison between standard LBFGS method and MBFGS using Non-Diagonal test function (2≤n≤100000)

Non-Diagonal	LBFGS $\rho = 1$		MBFGS	
Function	NOI	Function Value	NOI	Function Value
n=2	27	7.1D-16	21	2.2D-13
n=4	32	4.9D-15	34	5.5D-16
n=8	34	1.5D-13	25	4.4D-16
n=10	36	2.0D-15	27	3.3D-13
n=40	36	4.5D-11	30	2.3D-12
n=100	36	1.6D-15	34	1.8D-15
n=1000	40	7.3D-15	32	1.0D-15
n=5000	44	2.1D-14	27	9.2D-13
n=10000	44	2.8D-12	31	1.6D-19
n=100000	44	1.6D-13	36	5.1D-15
Total	373		297	

Table (4.10) B: Percentage performance of the new modification against the standard method using Non-Diagonal function

Tools	LBFGS $\rho = 1$	MBFGS
NOI	100%	79.62%

5. Numerical Results:

All routines are written in FORTRAN 77 and computational results are obtained on a Pentium IV machine. The required accuracy is set as 10^{-6}. That is, convergence is assumed if the following criterion is satisfied at the point x_k

$$\|g_k\| < 10^{-6} \times \max\{1, \|x_k\|\} \qquad (42)$$

where $\|.\|$ is the l_2 (Euclidean) norm.

A total of 10 standard functions have been chosen for evaluation purposes. Several of these functions are given by Gill and Murray [11]. Each function is tested with ten different dimensions, namely, n=2,4,8,10,40, 100, 1000, 5000, 10000, and 100000, m=5.

All test functions are tested with a single standard starting point.

NOI denotes the number of iteration.

The line search is based on backtracking, using quadratic and cubic modeling of $f(x)$ in the direction of search.

As can be seen from Tables 1-10 for most of the problems especially problems with large number of variables, the number of iterations required by the new MBFGS algorithm is less than the corresponding numbers required by LBFGS. This indicates that the new method is a better choice for solving large-scale optimization problems.

For the obtained numerical results, we have from tables (4.1)-(4.10) that taking NOI as the standard tool for comparison neglecting NOF because it depends up on NOI under the condition of using the cubic fitting technique as a linear search subprogram. The improvement percentage of the new method is between (28-91)%.

6: Conclusions:

We have presented a limited modified BFGS method for solving unconstrained large-scale optimization problems. The proposed algorithm generates quasi-Newton directions using a modified BFGS method suggested by Biggs [9]. This modified BFGS method is then extended to the limited memory version. In order to save storage, the modification is only applied to the last corrections. Numerical results indicate an overall improvement on the number of iteration and function/gradient evaluation.

References:

[1] C.G. Broyden, "The convergence of a class of double-rank minimization algorithms", J. Inst. Maths Applics., 6 (1970), 76-90.

[2] R. Fletcher, "A new approach to variable metric algorithm", Computer Journal,13, (1970), 392-399.

[3] D. Goldfarb,"A family of variable metric methods derived by variational means, Mathematics of Computations",24 (1970),23-26.

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

[5] L. Luksan, "Computational experience with known variable metric updates, Journal of Optimization Theory and Applications", 83 (1994), 27-47.

[6] J. Nocedal, "Updating quasi-Newton matrices with limited storage, Mathematics of Computation", 35 (1980), 773-782.

[7] J.E. Dennis, Jr. & R.B. Schnabel, "Numerical methods for unconstrained optimization and nonlinear equations", Prentice-Hill Inc., New Jersey, 1983.

[8] Malik Abu Hassan, Leong Wah June, & Mansor Monsi "Convergence of A Modified BFGS Method" Volume 22, Number 1, (2006), pp. 17–24.

[9] M.C. Biggs, "A note on minimization algorithms which make use of non-quadratic properties of the objective function", J. Inst. Maths Applics., 12 (1973), 337-338.

[10] Malik Hj. Abu Hassan, Mansor B. Monsi & Leong Wah June, "limited modified BFGS method for large-scale optimization, matematika", (2001), Jilid 17,bil.1, hlm. 15–23

[11] P.E. Gill & W. Murray, "Conjugate-gradient methods for large scale nonlinear optimization", Technical report SOL 79-15, Department of Operation Research, Stanford University, Stanford, 1979.

مقارنة بين خوارزميات LBFGS وخوارزمية المتري المتغير(VM) ذات القياس الحر في الأمثلية غير المقيدة

عمر بهاء الدين محمد

قسم الرياضيات ، كلية العلوم ، جامعة كويه ، اربيل ، العراق

(تاريخ الاستلام: ٢٨ / ٩ / ٢٠٠٩ ---- **تاريخ القبول:** ٢٣ / ٥ / ٢٠١٠ **)**

الملخص

في هذا البحث، تم المقارنة بين خوارزمية أل LBFGS المطورة من قبل نوسيدال (١٩٨٠) وخوارزمية المتري المتغير والتي سميت خوارزمية أل MBFGS.

الخوارزمية المقترحة هي المفضلة نسبة الى قلة الخزن المطلوبة وكذلك قدرتها على حل المسائل ذات القياس العالي تصل الى 10^6 من المتغيرات بنجاح بينما الطريقة الثانية تفشل في الحصول على نفس الدقة.