On Limited Memory Self-Scaling VM-Algorithms for Unconstrained Optimization

Abbas Y. Al-Bayati
Dept. of Mathematics
college of computer science & Mathematics
Mosul University

Omar B. Mohammed
Dept. of Mathematics
college of science
Koya University

Muna M. Mohammed Ali
Dept. of Mathematics
college of computer science & Mathematics
Mosul University

Received 29 / 03 / 2009 **Accepted** 05 / 05 / 2009

الملخص

في هذا البحث تم تحديد مقياس جديد لذاكرة الخـزن المحـدود لطريقـة الـ ـ (L-BFGS)، الغاية من البحث هي المقارنة مع طريقة الـ ـ (L-BFGS) القياسـية مـع m=3 وباستخدام عشرة دوال لا خطية مختلفة الأبعاد لاختبار المسائل. التحديث الجديد يشير إلى ان النتائج العددية فعالة جدا بالمقارنة مع الخوارزمية القياسية.

Abstract

In this paper, we have investigated a new scaling parameter in the standard memoryless L-BFGS algorithm. This new consideration is compared with the standard L-BFGS method under the assumption of the L-BFGS method with m=3, and by using ten nonlinear different dimensionality test problems. The new modification performs very effective numerical results compared with the standard algorithm.

1. Introduction

BFGS quasi-Newton methods have reliable and efficient for the unconstrained minimization of a smooth nonlinear function $f : R^n \to R$.

However, the need to store an n x n approximate Hessian has limited their application to problems with a small to medium number of variables.

For large n it is necessary to use methods that do not require the storage of a full n x n matrix. Sparse quasi-Newton updates can be applied if the Hessian has a significant number of zero entries (see, e.g., Powell and Toint [13], Fletcher [5].

In nonlinearly constrained optimization, other methods must be used. Such methods include conjugate-gradient methods, limited-memory quasi-Newton methods, and limited-memory reduced-Hessian quasi-Newton methods (see Gill,et al [7]).

1.1: Variable Metric Methods:

We have seen that in order to obtain a super linearly convergent method.

How can we do this without actually evaluating the Hessian matrix at every iteration? The answer was discovered by Dixon (1959), and was subsequently developed and popularized by (Fletcher and Powell [6]). It consists of starting with any approximation to the Hessian matrix, and at each iteration, update this matrix by incorporating the curvature of the problem measured along the step. If this update is done appropriately, one obtains some remarkably robust and efficient methods, called variable metric methods. they revolutionized nonlinear optimization by providing an alternative to Newton's method, which is too costly for many applications. There are many variable metric methods, but since 1970, the BFGS method has been generally considered to be the most effective. It is implemented in all major subroutine libraries and is currently being used to solve optimization problems arising in a wide spectrum of applications.

The theory of variable metric methods is beautiful. The more we study them, the more remarkable they seem. We now have a fairly good understanding of their properties. Much of this knowledge has been obtained recently, and we will discuss it in this section. We will see that the BFGS method has interesting self-correcting properties, which account for its robustness. We will also discuss some open questions that have resisted an answer for many years.

The BFGS method is a line search method. At the $k-th$ iteration, a symmetric and positive definite matrix B_k is given, and a search direction is computed by

$$d_k = -B_k^{-1} g_k \qquad (1)$$

The next iterate is Given by

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

where the step size (λ_k) satisfies the following Wolfe conditions:

$$f(x_k + \lambda_k d_k) \le f(x_k) + \sigma_1 \lambda_k g_k^T d_k \quad (3)$$

$$g\left(x_k + \lambda_k d_k\right)^T d_k \ge \sigma_2 g_k^T d_k \quad (4)$$

where $0 < \sigma_1 < \sigma_2 < 1$

It has been found that it is best to implement BFGS with a very loose line search: typical values for parameters in (3), (4) are $\sigma_1 = 10^{-4}$ and $\alpha_2 = 0.9$. The Hessian approximation is updated by:

$$B_{k+1} = B_K - \frac{B_K s_k s_k^T B_k}{s_k^T B_k s_k} + \frac{y_k y_k^T}{y_k^T s_k}. \quad (5)$$

s.t:

$$y_k = g_{k+1} - g_k, \qquad s_k = x_{k+1} - x_k. \quad (6)$$

Note that the two correction matrices on the right hand side of (5) have rank one. Therefore by the interlocking eigen value theorem Wilkinson, (1965), the first rank-one correction matrix, which is subtracted, decreases the eigen values. We will say that it "shift the eigen values to the left" on the other hand, the second rank one matrix. Which is added, shifts the eigen values to the right. There must be a balance between these eigen values shifts, for otherwise the Hessian approximation could either approach singularity or become arbitrarily large, causing a failure of the method.

A global convergence result for the BFGS method can be obtained by careful consideration of these eigen value shifts. This done by Powell [12], who uses the trace and the determinant to measure the effect of the two rank-one corrections on B_k. He is able to show that if f is convex, then for any positive definite starting matrix B_1 and any starting point x_1, the BFGS method gives lim inf $\|g_k\| = 0$. If in addition the sequence $\{x_k\}$ converges to a solution point at which the Hessian matrix is positive definite, then the rate of convergence is superlinear.

This analysis has been extended by Byrd, Nocedal and Yuan [4], to the restricted Broyden class of quasi-Newton methods in which (5) is replaced by

$$B_{k+1} = B_K - \frac{B_K s_k s_k^T B_k}{s_k^T B_k s_k} + \frac{y_k y_k^T}{y_k^T s_k} + \phi(s_k^T B_k s_k) v_k v_k^T \quad (7)$$

Where $\phi \in [0,1]$, and

$$v_k = \left[\frac{y_k}{y_k^T s_k} - \frac{B_k s_k}{(s_k^T B_k s_k)}\right].$$

The choice $\phi = 0$ gives rise to the BFGS update, whereas $\phi = 1$ defines the DFP method, the first variable metric method proposed by Davidon, Fletcher and Powell. Byrd, Nocedal and Yuan prove global and superlinear convergence on convex problems, for all methods in the restricted Broyden class, except for DFP. Their approach breaks down when $\phi = 1$, and leaves that case unresolved. Indeed the following question has remained unanswered since 1976, when Powell published his study on the BFGS method, [11].

2. The Limited Memory BFGS Method.

Quasi-Newton methods are a class of numerical methods that are similar to Newton's method except that the inverse of Hessian $(G(x_k))^{-1}$ is replaced by a n x n symmetric matrix H_k, which satisfies the quasi-Newton equation: (see [8]).

$$H_k y_{k-1} = s_{k-1}, \tag{8}$$

where

$$s_{k-1} = x_k - x_{k-1} = \lambda_{k-1} d_{k-1}, \qquad y_{k-1} = g_k - g_{k-1} \tag{9}$$

And $\lambda_{k-1} > 0$ is a step-length which satisfies some line search conditions. Assuming H_k nonsingular, we define $B_k = H_k^{-1}$. It is easy to see that the quasi-Newton step.

$$d_k = -H_k g_k \tag{10}$$

Is a stationary point of the following problem:

$$\min_{d \in R^n} \phi_k(d) = f(x_k) + d^T g_k + \frac{1}{2} d^T B_k d \tag{11}$$

Which is an approximation to problem $\min_{x \in R^n} f(x)$ near the current iterate x_k, since $\phi_k(d) \approx f(x_k + d)$ for small d. In fact, the definition of $\phi_k(d)$ in (11) implies that

$$\phi_k(0) = f(x_k), \nabla \phi_k(0) = g(x_k) \tag{12}$$

and the quasi-Newton condition (8) is equivalent to

$$\nabla \phi_k(x_{k-1} - x_k) = g(x_{k-1}). \tag{13}$$

Thus, $\phi_k(x - x_k)$ is a quadratic interpolation of $f(x)$ at x_k and x_{k-1}, satisfying conditions (11)-(12). The matrix B_k (or H_k) can be updated so that the quasi-Newton equation is satisfied.

One well known update formula is the BFGS formula which updates B_{k-1} from B_k, s_k and y_k in the following way:

$$B_{k+1} = B_K - \frac{B_K s_k s_k^T B_k}{s_k^T B_k s_k} + \frac{y_k y_k^T}{s_k^T y_k} \tag{14}$$

In Yuan [9], approximate function $\phi_k(d)$ in (11) is required to satisfy the interpolation condition.

$$\phi_k(x_{k-1} - x_k) = f(x_{k-1}) \tag{15}$$

instead of (13). This change was inspired from the fact that for one-dimensional problem, using (15) gives a slightly faster local convergence if we assume $\lambda_k = 1$ for all k. Equation (15) can be rewritten as

$$s_{k-1}^T B_k s_{k-1} = 2\left[f(x_{k-1}) - f(x_k) + s_{k-1}^T g_k\right]. \tag{16}$$

In order to satisfy (16), the BFGS formula is modified as follows:

$$B_{k+1} = B_K - \frac{B_K s_k s_k^T B_k}{s_k^T B_k s_k} + t_k \frac{y_k y_k^T}{s_k^T y_k} \tag{17}$$

where

$$t_k = \frac{2}{s_k^T y_k}\left[f(x_k - f(x_{k+1}) + s_k^T g_{k+1}\right]. \tag{18}$$

If H_{k+1} is the inverse of B_{k+1}, then

$$H_{k+1} = H_k + \frac{1}{s_k^T y_k}\left[\left(\alpha_k + \frac{y_k^T H_k y_k}{s_k^T y_k}\right)s_k s_k^T - s_k y_k^T H_k - H_k y_k s_k^T\right] \tag{19}$$

with

$$\alpha_k = \frac{1}{t_k} \tag{20}$$

Assume that B_k is positive definite and that $s_k^T y_k > 0$, B_{k+1} definite by (17) is positive definite if and only if $t_k > 0$. The inequality $t_k > 0$ is trivial if f is strictly convex, and it is also true if the step-length λ_k is chosen by an exact line search, which requires $s_k^T g_{k+1} = 0$. For a uniformly convex function, it can be easily shown that there exists a constant $\delta > 0$ such that $t_k \in [\delta, 2]$ for all k, and consequently global convergence proof of the BFGS method for convex functions with inexact line searches, which was given by Powell [12]. However, for a general nonlinear function f, inexact line searches do not imply the positivity of t_k, hence Yuan [15] truncated t_k to the interval [0.01,100], and showed that the global convergence of the modified BFGS algorithm is preserved for convex functions. If the objective function f is cubic along the line segment between x_{k-1} and x_k then we have the following relation

$$s_{k-1}^T G(x_k) s_{k-1} = 4 s_{k-1}^T g_k + 2 s_{k-1}^T g_{k+1} - 6\left[f(x_{k-1}) - f(x_k)\right] \tag{21}$$

By considering the Hermit interpolation on the line between x_{k-1} and x_k. Hence it is reasonable to require that the new approximate Hessian satisfies condition

$$s_{k-1}^T B_k s_{k-1} = 4 s_{k-1}^T g_k + 2 s_{k-1}^T g_{k+1} - 6\left[f(x_{k-1}) - f(x_k)\right] \tag{22}$$

Instead of (18). Biggs [2],[3] gives the update of (19) with the value t_k chosen so that (22) holds. The respected value of t_k is given by

$$t_k = \frac{6}{s_k^T y_k}\left[f(x_k - f(x_{k+1}) + s_k^T g_{k+1}\right] - 2 \tag{23}$$

For one-dimensional problems, Wang and Yuan [14] showed that (17) with (23) and without line searches (that is $\lambda_k = 1$ for all k) implies R-quadratic convergence, and expect some special cases (17) with (23) also give Q-convergence. It is well known that the convergence rate of secant method is $\left(1+\sqrt{5}\right)/2$ which is approximately 1.618 and less than 2.

The limited memory BFGS method is described by Nocedal [10], where it is called the SQN method. The user specifies the number m of BFGS 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 BFGS method. For $k > m$, H_k is obtained by applying m BFGS updates to H_0 using information from the m previous iterations. The method uses the inverse BFGS formula in the form

$$H_{k+1} = V_k^T H_k V_k + \rho_k s_k s_k^T, \tag{24}$$

Where

$$\rho_k = \frac{1}{y_k^T s_k}, \qquad \mathrm{V_k} = I - \rho_k y_k s_k^T. \tag{25}$$

(see [3]).

2-1: Non-Convex Functions

All the results for the BFGS method discussed so far depend on the assumption that the objective function f is convex. At present, few results are available for the case in which f is a more general nonlinear function. Even though the numerical experience of many years suggests that the BFGS method always converges to a solution point, this has not been proved.

Consider the BFGS method with a line search satisfying the Wolfe conditions (3), (4). Assume that f is twice continuously differentiable and bounded below. Do the iterates satisfy lim inf $\|g_k\| = 0$, for any starting point x_1 and any positive definite starting matrix B_1?

This is one of the most fundamental questions in the theory of unconstrained optimization, for BFGS is perhaps the most commonly used method for solving nonlinear optimization problems. It is remarkable that the answer to this question has not yet been found. Nobody has been able to construct an example in which the BFGS method fails, and the most general result available to us, due to (Powell [12]).

2-2: Outlines of the limited memory BFGS algorithm

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

step 2: Compute

$$d_k = -H_k g_k \tag{26}$$

and

$$x_{k+1} = x_k + \lambda_k d_k \tag{27}$$

where λ_k satisfies:

$$f(x_k + \lambda\alpha_k d_k) < f(x_k) + \beta' \lambda_k g_k^T, \quad (28)$$

$$g(x_k + \lambda_k d_k)^T d_k \geq \beta g_k^T d_k. \quad (29)$$

(the step-size $\lambda = 1$ k is tried first).

step 3: Let $m = \min\{k, m-1\}$. Update H_0 for $m+1$ times by using the pairs $\{y_i, s_i\}_{j=k-m}^{k}$, i.e. let

$$\begin{aligned} H_{k+1} &= (V_k^T \ldots V_{k-m}^T) H_0 (V_{k-m} \ldots V_k) \\ &+ \rho_{\text{k-m}} (V_k^T \ldots V_{k-m+1}^T) s_{k-m} s_{k-m}^T (V_{k-m+1} \ldots V_k) \\ &+ \rho_{\text{k-m+1}} (V_k^T \ldots V_{k-m+2}^T) s_{k-m+1} s_{k-m+1}^T (V_{k-m+2} \ldots V_k) \\ &+ \rho_{\text{k}} s_k s_k^T \end{aligned} \quad (30)$$

step 4: If $\|g_{k+1}\| < \varepsilon$ then stop, otherwise, put k = k+1 and Goto step (2).

3. Derivation of a new Scaling parameter

From section (2) above we have observed that taking $\alpha_k = 1$ from (19) yields the standard BFGS method, Now, taking the scalar $\alpha_k = \frac{y_k^T H_k y_k}{y_k^T s_k}$ which was known as Al-Bayati [1] parameter, with our consideration that for the purpose of the storage of the matrix H, we considered that $H_k = I_k$, so we have obtained a new scalar parameter, namely $\alpha_k = \frac{y_k^T y_k}{y_k^T s_k}$, because this quantity does not need the calculation of the matrix H_k at every iteration.

3-1: Suppose that f is differentiable and bounded below. Consider the BFGS method with a line search satisfying the Wolfe conditions (3), (4). Then lim inf $\|g_k\| = 0$ for any starting point x_1 and any positive definite starting matrix B_1 if the parameter

$$\left\{ \frac{y_k^T y_k}{y_k^T s_k} \right\} \quad (31)$$

is bounded above for all k**.**

3-2: Outlines of the new algorithm

step 1: Choose x_0 as initial point.

step 2: Let $\varepsilon_0 > 0$.

step 3: Put k=0, repeat.

step 4: Compute $d_k = -H_k g_k$, and $x_{k+1} = x_k + \lambda_k d_k$,

where λ_k satisfies wolfe conditions (3),(4).

step 5: Compute $s_k = x_{k+1} - x_k$, $y_k = g_{k+1} - g_k$.

step 6: Compute H_k from

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

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

and

$$\begin{aligned} H_{k+1} &= (V_k^T \ldots V_{k-3}^T) H_0 (V_{k-3} \ldots V_k) \\ &+ \rho_{k-3} (V_k^T \ldots V_{k-2}^T) s_{k-3} s_{k-3}^T (V_{k-2} \ldots V_k) \\ &+ \rho_{k-2} (V_k^T \ldots V_{k-1}^T) s_{k-2} s_{k-2}^T (V_{k-1} \ldots V_k) \\ &+ \rho_k s_k s_k^T \end{aligned}$$

step 7: If $\|g_{k+1}\| < \varepsilon$ then stop, otherwise, put k = k+1 and Go to step (4).

4. Computational Results:

In this section, we present and discus some numerical experiments that were conducted in order to test the performance of limited memory Quasi-Newton methods for unconstrained optimization using the standard BFGS formulae again using modified BFGS update.

The algorithms used for limited memory methods are form L-BFGS, which provides the line search strategy for calculating global step. The line search is based on backtracking, using quadratic and cubic modeling of $f(x)$ in the direction of search.

Ten test functions, with variable dimensions, have been chosen from literature of optimization. The description of these test problems can be found, for instance, in More et al. [9]. Each function is tested with seven different dimensions, namely n=2,4,10,100,500,1000 and 10000, m=3. All test functions are tested with a single standard starting point.

All algorithms are implemented in FORTRAN. The runs were performed with a double precision arithmetic, for which the unit round off is approximately 10^{-16}. In all cases, convergence is assumed if

$$\|g_k\| < 10^{-5} \qquad (32)$$

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 (13 - 41)%.

Table (4.1) A

(Comparison between standard L-BFGS method and modified L-BFGS using Dixon test function (2≤n≤10000)

Dixon	BFGS $\alpha_k = 1$		$\alpha_k = \frac{y_k^T y_k}{y_k^T s_k}$	
Function	NOI	Function Value	NOI	Function Value
n=2 M=3	9	1.E-3	8	2.E-17
n=4 M=3	21	1.E-11	15	2.E-16
n=10 M=3	58	6.E-11	32	8.E-11
n=100 M=3	282	5.E-1	32	1.E-9
n=500 M=3	315	5.E-1	34	1.E-4
n=1000 M=3	310	5.E-1	27	1.E-8
n=10000 M=3	307	5.E-1	708	5.E-1
Total	1302		856	

Table (4.1) B

Percentage performance of the improvement of the new modification against the standard method using Dixon function

Tools	BFGS	New Scale	Improvement percentage
NOI	100%	65.75%	35%

Table (4.2) A

(Comparison between standard L-BFGS method and modified L-BFGS using Wood test function (2≤n≤10000)

Wood	BFGS $\alpha_k = 1$		$\alpha_k = \frac{y_k^T y_k}{y_k^T s_k}$	
Function	NOI	Function Value	NOI	Function Value
n=2 M=3	18	9.E+3	17	9.E+3
n=4 M=3	112	1.E-13	70	1.E-13
n=10 M=3	253	9.E+3	205	9.E+3
n=100 M=3	102	1.E-11	77	4.E-12
n=500 M=3	115	7.E-11	80	2.E-14
n=1000 M=3	122	3.E-9	69	2.E-12
n=10000 M=3	98	2.E-8	100	5.E-12
Total	820		618	

Table (4.2) B

Percentage performance of the improvement of the new modification against the standard method using Wood function

Tools	BFGS	New Scale	Improvement percentage
NOI	100%	75.37%	25%

Table (4.3) A
(Comparison between standard L-BFGS method and modified L-BFGS using Shallow test function (2≤n≤10000)

Shallow	BFGS $\alpha_k = 1$		$\alpha_k = \frac{y_k^T y_k}{y_k^T s_k}$	
Function	NOI	Function Value	NOI	Function Value
n=2 M=3	13	5.E-14	10	1.E-15
n=4 M=3	14	1.E-13	11	5.E-15
n=10 M=3	13	3.E-15	10	6.E-13
n=100 M=3	15	1.E-10	14	8.E-11
n=500 M=3	13	6.E-14	10	2.E-9
n=1000 M=3	13	2.E-10	12	2.E-12
n=10000 M=3	13	7.E-9	11	1.E-13
Total	94		78	

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

Tools	BFGS	New Scale	Improvement percentage
NOI	100%	82.98%	18%

Table (4.4) A
(Comparison between standard L-BFGS method and modified L-BFGS using Cubic test function (2≤n≤10000)

Cubic	BFGS $\alpha_k = 1$		$\alpha_k = \frac{y_k^T y_k}{y_k^T s_k}$	
Function	NOI	Function Value	NOI	Function Value
N=2 M=3	36	3.E-14	25	8.E-10
N=4 M=3	34	6.E-16	19	2.E-10
N=10 M=3	38	3.E-16	24	1.E-14
N=100 M=3	37	3.E-12	23	6.E-13
N=500 M=3	38	4.E-11	21	1.E-11
N=1000 M=3	42	1.E-15	24	7.E-15
N=10000 M=3	36	2.E-15	18	4.E-11
Total	261		154	

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

Tools	BFGS	New Scale	Improvement percentage
NOI	100%	59.0%	41%

Table (4.5) A
(Comparison between standard L-BFGS method and modified L-BFGS using Rosen test function (2≤n≤10000)

Rosen	BFGS $\alpha_k = 1$		$\alpha_k = \frac{y_k^T y_k}{y_k^T s_k}$	
Function	NOI	Function Value	NOI	Function Value
N=2 M=3	38	5.E-16	33	4.E-13
n=4 M=3	38	1.E-15	27	3.E-18
n=10 M=3	36	2.E-13	26	1.E-11
n=100 M=3	32	3.E-15	37	8.E-13
n=500 M=3	35	1.E-11	26	6.E-19
n=1000 M=3	34	1.E-10	27	1.E-14
n=10000 M=3	34	2.E-16	34	2.E-9
Total	247		210	

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

Tools	BFGS	New Scale	Improvement percentage
NOI	100%	85.02%	15%

Table (4.6) A
(Comparison between standard L-BFGS method and modified L-BFGS using Non-Diagonal test function (2≤n≤10000)

Non-Diagonal	BFGS $\alpha_k = 1$		$\alpha_k = \frac{y_k^T y_k}{y_k^T s_k}$	
Function	NOI	Function Value	NOI	Function Value
n=2 M=3	29	1.E-14	31	2.E-15
n=4 M=3	34	5.E-18	32	1.E-20
n=10 M=3	37	3.E-16	30	1.E-17
n=100 M=3	37	6.E-12	36	2.E-15
n=500 M=3	41	3.E-14	29	7.E-16
n=1000 M=3	39	2.E-14	25	1.E-12
n=10000 M=3	44	8.E-16	35	5.E-16
Total	261		218	

Table (4.6) B
Percentage performance of the improvement of the new modification against the standard method using non-Diagonal function

Tools	BFGS	New Scale	Improvement percentage
NOI	100%	83.52%	17%

Table (4.7) A
(Comparison between standard L-BFGS method and modified L-BFGS using Fox test function (2≤n≤10000)

Fox	BFGS $\alpha_k = 1$		$\alpha_k = \frac{y_k^T y_k}{y_k^T s_k}$	
Function	NOI	Function Value	NOI	Function Value
n=2 M=3	11	-5.E-1	10	-5.E-1
n=4 M=3	11	-1.E0	9	-1.E0
n=10 M=3	10	-2.E0	9	-2.E0
n=100 M=3	12	-2.E+1	10	-2.E+1
n=500 M=3	11	-1.E+2	10	-1.E+2
n=1000 M=3	12	-2.E+2	10	-2.E+2
n=10000 M=3	11	-2.E+3	10	-2.E+3
Total	78		68	

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

Tools	BFGS	New Scale	Improvement percentage
NOI	100%	87.18%	13%

Table (4.8) A
(Comparison between standard L-BFGS method and modified L-BFGS using Pen(1) test function (2≤n≤10000)

Start	BFGS $\alpha_k = 1$		$\alpha_k = \frac{y_k^T y_k}{y_k^T s_k}$	
Function	NOI	Function Value	NOI	Function Value
n=2 M=3	3	4.E-32	3	1.E-32
n=4 M=3	3	8.E-32	3	2.E-32
n=10 M=3	7	5.E-14	4	6.E-32
n=100 M=3	6	1.E-13	4	9.E-30
n=500 M=3	7	6.E-18	4	4.E-28
n=1000 M=3	7	2.E-17	4	2.E-26
n=10000 M=3	7	1.E-16	4	2.E-27
Total	40		26	

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

Tools	BFGS	New Scale	Improvement percentage
NOI	100%	65%	35%

Table (4.9) A
(Comparison between standard L-BFGS method and modified L-BFGS using Powell test function (2≤n≤10000)

Powell	BFGS $\alpha_k = 1$		$\alpha_k = \frac{y_k^T y_k}{y_k^T s_k}$	
Function	NOI	Function Value	NOI	Function Value
n=2 M=3	38	8.E-18	31	2. E-18
n=4 M=3	38	4. E-15	26	1. E-13
n=10 M=3	35	2. E-13	27	4. E-12
n=100 M=3	33	4. E-19	34	1. E-15
n=500 M=3	35	1. E-11	24	1. E-12
n=1000 M=3	35	2. E-16	30	1. E-11
n=10000 M=3	34	2. E-16	35	4. E-13
Total	248		207	

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

Tools	BFGS	New Scale	Improvement percentage
NOI	100%	83.48%	17%

Table (4.10) A
(Comparison between standard L-BFGS method and modified L-BFGS using Pen(2) test function (2≤n≤10000)

Pen(2)	BFGS $\alpha_k = 1$		$\alpha_k = \frac{y_k^T y_k}{y_k^T s_k}$	
Function	NOI	Function Value	NOI	Function Value
n=2 M=3	3	5.E-6	4	5.E-6
n=4 M=3	3	1. E-5	2	1. E-5
n=10 M=3	7	2. E-5	4	2. E-5
n=100 M=3	8	2. E-4	4	2. E-4
n=500 M=3	8	1. E-3	5	1. E-3
n=1000 M=3	8	2. E-3	6	2. E-3
n=10000 M=3	8	2. E-3	5	2. E-3
Total	45		30	

Table (4.10) B
Percentage performance of the improvement of the new modification against the standard method using Pen(2) function

Tools	BFGS	New Scale	Improvement percentage
NOI	100%	66.66%	34%

5. Conclusions:

We have attempted, in this paper, to develop a numerical procedure for solving large-scale unconstrained optimization problem that are based on different technique of approximating the objective function. We have applied BFGS updated and the new scale in the limited memory scheme replacing the standard BFGS update.

We have tested these algorithms on a set of standard test functions from More et al. [14]. Our test results show that on the set of problems we tried, our partially modified L-BFGS methods require fewer iterations, gradient evaluations and the minimum value is less than L-BFGS by Nocedal [13].

Numerical tests also suggest that these partially modified L-BFGS algorithms are more superior than the standard L-BFGS algorithms. Thus for large problems where space limitations do not preclude using the full quasi-Newton updates, these methods are recommended.

REFERENCES

1) Al- Bayati, A. Y. (A new family of self-scaling VM algorithms for unconstrained optimization) J. Educ. and Sci, Mosul, IRAQ, 12: 25-54, (1991).

2) 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).

3) 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)

4) Byrd, J. Nocedal and Y. Yuan, Global convergence of a class of quasi-Newton methods on convex problems, SIAM J. Numer. Anal.24. 1171-1190, (1987).

5) Fletcher R., An optimal positive definite update for spare Hessian matrices SIAM J. Optim., 5:PP.192-218, (1995).

6) Fletcher R. and Powell M.J.D., a rapidly convergent descent method for minimization, Comput. J. 6:163-168, (1963).

7) Gill. P and Michael W. Leonard, Limited-Memory Reduced-Hessian Methods For Large-Scale Unconstrained Optimization, AMS, (2000).

8) Leong Wah June AND Malik Abu Hassan, Modification of the limited memory BFGS algorithm for large-scale nonlinear optimization, Math. J. Okayama Univ. 47:175-188, (2005).

9) More, J. J., Garbow, B.S. and Hillstrom, K. E. Testing unconstrained optimization software. ACM Transactions on Mathematical Softwaew, 7:17-41, (1981).

10) Nocedal, J. Updating quasi-Newton matrices with limited storage. Mathematics of coputation, 35:773-782, (1980).

11) J. Nocedal, Theory of algorithms for unconstrained optimization, Acta Numerica, 1: pp. 199-242 (1991)

12) Powell, M. J. D, Some global convergence properties of a variable metric algorithm for minimization without exact line searches, in R.W. Cottle and C.E. Lemke, eds., Nonlinear Programming SIAM-AMS Proceeding, SIAM Publications, 9:53-72, (1976).

13) Powell M. J. D. and Toint P. L., On the estimation of sparse Hessian matrices, SIAM J. Numer. Anal., 16:PP.1060-1074 (1979).

14) Wang, H. J. and Yuan, Y. A quadratic convergence method for one-dimensional optimization. Chinese Journal of Operations Research, 11:12-10, (1992).

15) Yuan, Y. A modified BFGS algorithm for unconstrained optimization. IMA Journal Numerical Analysis, 11:325-332, (1991).