Two Modified QN-Algorithms for Solving Unconstrained Optimization Problems

Abbas Y. Al-Bayati  
Basim A. Hassan

profabbasalbayati@yahoo.com  
basimabas39@gmail.com

College of Computer Sciences and Mathematics  
University of Mosul, Mosul, Iraq

Received on: 20/04/2011  
Accepted on: 03/10/2011

ABSTRACT

This paper presents two modified Quasi-Newton algorithms which are designed for solving nonlinear unconstrained optimization problems. These algorithms are based on different techniques namely: Quasi-Newton conditions on quadratic and non-quadratic objective functions. Experimental results indicate that the new proposed algorithms are more efficient than the Yuan and Biggs-algorithms.

Keywords: Quasi-Newton method, Quasi-Newton condition, Experimental results.

1. Introduction.

Variable Metric or precisely Quasi-Newton (QN)-algorithms are used to solve a class of numerical methods of the following unconstrained optimization problem:

\[
\min \left\{ f(x) \mid x \in \mathbb{R}^n \right\} \quad \text{subject to} \quad \text{Constraints}
\]

where \( f \) is a smooth function of \( n \) variables [10]. We recall that these types of methods are iterative. Starting with an initial point \( x_0 \in \mathbb{R}^n \), they generate a sequence \( x_k \in \mathbb{R}^n \), by the process

\[
x_{k+1} = x_k + \alpha_k d_k,
\]

where \( d_k \) is a direction vector and the step size \( \alpha_k \) is chosen in such a way that \( \alpha_k > 0 \) and satisfies the Wolfe (W) conditions

\[
f(x_k + \alpha_k d_k) \leq f(x_k) + \delta_1 \alpha_k d_k^T g_k
\]

\[
g(x_k + \alpha_k d_k)^T d_k \geq \delta_2 \alpha_k d_k^T g_k
\]

with \( \delta_1 < 1/2 \) and \( \delta_1 < \delta_2 < 1 \), where \( f_k = f(x_k), \ g_k = g(x_k), d_k \) is the gradient of \( f \) evaluated at the current iterate \( x_k \) [9]. The search direction is calculated by:
\[ d_k = -B_k^{-1} g_k \] 
where \( B_k^{-1} \) is a symmetric positive definite matrix and satisfying the QN- equation

\[ B_{k+1} v_k = y_k, \]  
where \( v_k = x_{k+1} - x_k \) and \( y_k = g_{k+1} - g_k \) see [9, 10]. The search direction \( d_k \) in (5) is the solution of the following quadratic sub problem

\[ \min_{d \in \mathbb{R}^n} \mathcal{G}(d) = f(x_k) + d^T g(x_k) + \frac{1}{2} d^T B_k d, \]  

which is an approximation to problem (1) near the current iterate \( x_k \), since \( \mathcal{G}(d) \approx f(x_k + d) \), for small \( d \). In fact, the definition of \( \mathcal{G}(\cdot) \) in (7) implies that

\[ \mathcal{G}_k(0) = f(x_k), \]  
\[ \nabla \mathcal{G}_k(0) = \nabla f(x_k), \]  
and condition (6) is equivalent to

\[ \nabla \mathcal{G}_k(x_{k-1} - x_k) = \nabla f(x_{k-1}) = g(x_{k-1}). \]  

Thus, \( \nabla \mathcal{G}_k(x_{k-1} - x_k) \) is a quadratic interpolation of \( f_k \) at \( x_k \) and \( x_{k-1} \), satisfying condition (8−10). Davidon [4] introduced 'conic models' where a non-quadratic function \( \nabla \mathcal{G}_k(d) \) is constructed and \( \nabla \mathcal{G}_k(d) \) is satisfied at condition (8−10) and the interpolation condition

\[ \mathcal{G}_k(x_{k-1} - x_k) = f(x_{k-1}) \]  
more details can be found in [10].

In section (2), a modified Biggs's [1] and [2] update and Yuan's [10] update which are based on the simple idea of approximation the objective function by different techniques are induced. Finally, in section (3) numerical results with a brief discussion are presented.

2. Two Modified QN-Methods.

The BFGS algorithm for unconstrained optimization problem (1) uses the search direction(5), and the matrices \( B_k \) are updated by the BFGS formula as :

\[ B_{k+1}^{BFGS} = B_k - \frac{B_k v_k v_k^T B_k}{v_k^T B_k v_k} + \frac{y_k y_k^T}{y_k^T v_k}, \]  
which satisfied the QN equation (6). If \( H_{k+1} \) is the inverse of \( B_{k+1} \), then

\[ H_{k+1}^{BFGS} = H_k - \frac{H_k y_k v_k^T + v_k y_k^T H_k + v_k v_k^T}{y_k^T v_k} \left[ 1 + \frac{y_k^T H_k y_k}{y_k^T v_k} \right]. \]  
The BFGS method is one of the most efficient methods for solving the unconstrained optimization problem (1). More details can be found in Fletcher [5].

In [7] and [10], approximate function \( \mathcal{G}_k(d) \) in (7) is required to satisfy the interpolation condition (11) instead of (10) This change was inspired from the fact that for one dimensional problem, using (11) gives a slightly faster local convergence if we assume \( \alpha_k = 1 \) for all \( k \). Equation (11) can be rewritten as

\[ v_k^T B_{k+1} v_k = 2 \left[ f(x_k) - f(x_{k+1}) + v_k^T g_{k+1} \right]. \]  

In order to satisfy (14), the BFGS formula is modified as follows :

\[ B_{k+1} = B_k - \frac{B_k v_k v_k^T B_k}{v_k^T B_k v_k} + t_k \frac{y_k v_k^T}{y_k^T v_k}, \]
Two Modified QN-Algorithms for Solving Unconstrained Optimization Problems

where \( t_k^{Update} = \frac{2}{v_k^T y_k} \left[ f(x_k) - f(x_{k+1}) + v_k^T g_{k+1} \right] \) \ldots \ldots (16)

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

\[
H_{k+1} = H_k - \frac{H_k y_k y_k^T + v_k v_k^T H_k}{y_k^T v_k + v_k^T y_k} \left[ \rho_k + \frac{y_k^T H_k y_k}{y_k^T v_k + v_k^T y_k} \right]
\] \ldots \ldots (17)

with \( \rho_k = 1/t_k \).

However, condition (16) may be modified further to give:

\[
t_k^{Modified} = \frac{1}{v_k^T y_k} \left[ 2(f(x_i) - f(x_{i+1})) + 2v_i^T g_{i+1} \right]
\]

\[
= \frac{1}{v_k^T y_k} \left[ 2(f(x_i) - f(x_{i+1})) + v_i^T g_{i+1} + v_i^T g_{i+1} - v_i^T g_i \right]
\] \ldots \ldots (18)

\[
= \frac{1}{v_k^T y_k} \left[ 2(f(x_i) - f(x_{i+1})) + v_i^T (g_{i+1} - g_i) + v_i^T (g_{i+1} + g_i) \right]
\]

\[
= 2(f(x_i) - f(x_{i+1})) + v_i^T (g_{i+1} + g_i) - \frac{v_i^T y_k}{v_k^T y_k}
\]

In [6] and [7] if the objective function \( f \) is cubic along the line segment between \( x_{k-1} \) and \( x_k \), then we have the following relation:

\[
v_k^T f_k v_{k-1} = 4v_k^T g_k + 2v_k^T g_{k+1} - 6[f(x_{k-1}) - f(x_k)]
\] \ldots \ldots (19)

by considering the Hermit interpolation on the line between \( x_{k-1} \) and \( x_k \). Hence, it is reasonable to require that the approximate Hessian satisfies the condition of

\[
v_k^T B_{k+1} v_k = 4v_k^T g_k + 2v_k^T g_{k+1} - 6[f(x_k) - f(x_{k+1})]
\] \ldots \ldots (20)

Instead of (16).

Biggs [1] and [2] give the update of (17) with the value \( t_k \) chosen so that (20) holds. The respected value of \( t_k \) is given by

\[
t_k^{Biggs} = \frac{6}{v_k^T y_k} \left[ f(x_k) - f(x_{k+1}) \right] + v_k^T g_{k+1} - 2 \]
\] \ldots \ldots (21)

Thus we can obtain another modified parameter from (21) by considering the following relation:

\[
t_k^{Modified} = \frac{3}{v_k^T y_k} \left[ 2f(x_k) - f(x_{k+1}) \right] + v_k^T g_{k+1} - 2
\]

\[
= \frac{3}{v_k^T y_k} \left[ 2f(x_k) - f(x_{k+1}) \right] + v_k^T g_{k+1} + v_k^T g_{k+1} - v_k^T g_i - 2
\]

\[
= \frac{3}{v_k^T y_k} \left[ 2f(x_k) - f(x_{k+1}) \right] + v_k^T (g_{k+1} - g_i) + v_k^T (g_{k+1} + g_i) - 2
\] \ldots \ldots (22)

\[
= \frac{3}{v_k^T y_k} \left[ 2f(x_k) - f(x_{k+1}) \right] + v_k^T y_k + v_k^T (g_{k+1} + g_i) - 2
\]

\[
= \frac{6(f(x_k) - f(x_{k+1})) + 3(v_k^T (g_{k+1} + g_i) + v_k^T y_k)}{v_k^T y_k} - 2
\]

2.1 Two Modified QN-Algorithms

The outline of the modified QN-algorithm is as follows:
Step 0 : Choose an initial point \( x_0 \in \mathbb{R}^n \) and an initial positive definite matrix \( H_0 = I \), \( \varepsilon = 1 \times 10^{-4} \), set \( k = 1 \).

Step 1 : If \( \|g_{k+1}\| \leq \varepsilon \), stop.

Step 2 : Solve \( d_k = -H_k g_k \) to obtain a search direction \( d_k \).

Step 3 : Generate a new iteration point by \( x_{k+1} = x_k + \alpha_k d_k \) (Use Wolfe line search technique to compute the parameter \( \alpha_k \)) and calculate the new updating formula (17-18) and (17-22), if \( \rho_k < 0 \) or \( \rho_k > 1 \) then \( \rho_k = 1 \).

Step 4 : Set \( k = k + 1 \) and go to Step 1.

Assume that \( B_k \) is positive definite and that \( v_k^T y_k > 0 \), \( B_{k+1} \) defined by (17) is positive definite if and only if \( t_k > 0 \). However, for a general nonlinear function \( f \), inexact line searches do not imply the positivity of \( t_k \) and \( \rho_k = 1/t_k \), hence \( 0 < \rho_k < 1 \).

By slightly modifying the proof in Powell [8], it can be shown that algorithm 2.1 converges globally for convex objective functions with inexact line searches. Assume \( x_k \) converges to a strict local minimum \( x^* \) where \( \nabla f(x^*) = 0 \) and \( \nabla^2 f(x^*) \) is positive definite, and that \( f(x) \) is twice continuously differentiable. Then, it can be proved that

\[
\lim_{k \to \infty} t_k = 1.
\]

thus, it is reasonable to hope that local super linear convergence of the BFGS algorithm can be extended to the modified algorithm where updating formula (17) is used. Details of local analyses of the BFGS algorithm can be found in Dennis and More [3].

3. Numerical Results.

In this paper, we have proposed two versions of a modified VM-method, for solving unconstrained minimization nonlinear problems. The computational experiments show that the modified approaches given in this paper are successful. We claim that the two modified (1) and (2) are better than the Yuan and Biggs methods. We have selected (8) large scale unconstrained optimization problems in extended or generalized form, for each test function, we have considered numerical experiment with the number of variables, \( n = 100, 500 \) and \( 1000 \). The programs were written in Fortran 90. The same line search was employed in each algorithm, this was the cubic interpolation technique which satisfies the conditions (3) and (4) for convex optimization with \( \delta_1 = 0.0001 \) and \( \delta_2 = 0.9 \). We tabulate for comparison of these algorithms, the number of function evaluations (NOF) and the number of iterations (NOI).
### Table (3.1)

| Problem | n  | NOI | NOF | NOI | NOF |
|---------|----|-----|-----|-----|-----|
| 1       | 100| 63  | 164 | 34  | 93  |
|         | 500| 65  | 161 | 44  | 113 |
|         | 1000| 103 | 260 | 89  | 224 |
| 2       | 100| 250 | 709 | 237 | 663 |
|         | 500| 748 | 2013| 761 | 2038|
|         | 1000| 1208| 3030| 1191| 2933|
| 3       | 100| 27  | 85  | 30  | 92  |
|         | 500| 31  | 88  | 31  | 92  |
|         | 1000| 29  | 88  | 33  | 98  |
| 4       | 100| 11  | 45  | 9   | 34  |
|         | 500| 13  | 56  | 10  | 42  |
|         | 1000| 13  | 56  | 10  | 42  |
| 5       | 100| 224 | 605 | 240 | 646 |
|         | 500| 590 | 1483| 594 | 1444|
|         | 1000| 807 | 1845| 542 | 1199|
| 6       | 100| 8   | 21  | 8   | 21  |
|         | 500| 9   | 23  | 9   | 23  |
|         | 1000| 9   | 23  | 9   | 23  |
| 7       | 100| 6   | 20  | 6   | 20  |
|         | 500| 6   | 20  | 6   | 20  |
|         | 1000| 6   | 20  | 6   | 20  |
| 8       | 100| 62  | 135 | 62  | 125 |
|         | 500| 72  | 145 | 72  | 145 |
|         | 1000| 118 | 289 | 82  | 165 |
| Total   |   | 4478| 11384| 4115| 10315|

### Table (3.2)

| Problem | n  | NOI | NOF | NOI | NOF |
|---------|----|-----|-----|-----|-----|
| 1       | 100| 42  | 101 | 41  | 103 |
|         | 500| 43  | 108 | 41  | 103 |
|         | 1000| 124 | 305 | 109 | 269 |
| 2       | 100| 253 | 725 | 260 | 741 |
|         | 500| 846 | 2524| 823 | 2439|
|         | 1000| 1347| 3711| 1341| 3677|
| 3       | 100| 24  | 76  | 31  | 92  |
|         | 500| 35  | 106 | 27  | 81  |
|         | 1000| 28  | 82  | 35  | 109 |
| 4       | 100| 11  | 45  | 11  | 45  |
|         | 500| 22  | 93  | 15  | 66  |
|         | 1000| 21  | 83  | 19  | 79  |
| 5       | 100| 250 | 707 | 268 | 737 |
|         | 500| 875 | 2524| 823 | 2439|
|         | 1000| 1002| 2449| 995 | 2486|
| 6       | 100| 9   | 27  | 9   | 27  |
|         | 500| 9   | 27  | 9   | 27  |
|         | 1000| 9   | 27  | 9   | 27  |
| 7       | 100| 6   | 21  | 6   | 21  |
Table (3.1) gives a comparison between the Yuan-algorithm and the modified Yuan–algorithm for convex optimization, this table indicates that the modified algorithm saves 8% NOI and 9% NOF, overall against the standard Yuan–algorithm, especially for our selected test problems. The Percentage Performance of the improvements of the Table (3.1) is given by the following table:

Table (3.3) Relative Efficiency of the Yuan and the Modified Yuan.

| Tools | Yuan | Modified Yuan |
|-------|------|--------------|
| NOI   | 100 %| 91.89 %      |
| NOF   | 100 %| 90.60 %      |

However, Table (3.2) gives a comparison between the Biggs-algorithm and the modified Biggs-algorithm for convex optimization, this table indicates that the modified algorithm saves 0.5% NOI and 1.5% NOF, overall against the standard Biggs algorithm, especially for our selected test problems. The Percentage Performance of the improvements of the Table (3.2) is given by the following table:

Table (3.4) Relative Efficiency of the Biggs and Modified Biggs.

| Tools | Biggs | Modified Biggs |
|-------|-------|----------------|
| NOI   | 100 % | 99.82 %        |
| NOF   | 100 % | 98.22 %        |
REFERENCES

[1] Biggs, M.C. (1971), Minimization algorithms making use of non-quadratic properties of the objective function. Journal of the Institute of Mathematics and Its Application, 8, pp. 315-327.

[2] Biggs, M.C. (1973), A note on Minimization algorithms making use of non-quadratic properties of the objective function. Journal of the Institute of Mathematics and Its Application, 12, pp. 337-338.

[3] Dennis, J.E. and More, J.J., (1977), Quasi-Newton methods, motivation and theory, SIAM Review, 19, pp.46-89.

[4] Davidon, W.C., (1980), Conic approximations and collinear scaling for optimizers, SIAM Journal Numerical Anal., 17, pp. 268-281.

[5] Fletcher, R. (1987), Practical Methods of Optimization John Wiley and Sons, Chi Chester (New York).

[6] Hassan, M.A., June, L.W and Monsi, M. (2005), Modified of the limited method BFGS algorithm for large-scale nonlinear optimization. Mathematics Journal Okayama Univ., 47, pp. 175-188.

[7] Hassan, M.A., June, L.W and Monsi, M. (2006), Convergence of the Modified BFGS Method, MATEMATIKA, 22, pp.17-24.

[8] Powell M. J. (1976), Some global convergence properties of a variable metric algorithm for minimization without exact searches, Nonlinear Programming SIAM-AMS Proceeding, 9, pp.53-72.

[9] Vlcek, J. and Luksan, L., (2004), A additional properties of variable metric methods , Technical report, ICS AS CR, 899, pp. 1-15

[10] Yuan, Y. (1991), A modified BFGS algorithm for unconstrained optimization. IMA Journal Numerical Analysis, 11, pp. 325-332.
Appendix

1. Generalized Powell function:
\[
f(x) = \sum_{i=1}^{n/4} (x_{4i-3} - 10x_{4i-2})^2 + 5(x_{4i-1} - x_i)^2 + (x_{4i-2} - 2x_i)^2 + 10(x_{4i-3} - x_i)^4 + (x_{4i-2} - 2x_i - x_{4i-3})^2
\]

Starting point: (3,1,0,1,...........,....)^T

2. Generalized Wood function:
\[
f(x) = \sum_{i=1}^{n/4} 4(x_{4i-2} - x_{4i-3})^2 + (1 - x_{4i-1})^2 + 90(x_{4i} - x_{4i-1})^2 + (1 - x_{4i-1})^2 + 10.1((x_{4i-2} - 1)^2 + (x_{4i} - 1)^2 + 19.8((x_{4i-2} - 1) + (x_{4i} - 1))
\]

Starting point: (-3,-1,-3,-1,............,...)^T

3. Miele function:
\[
f(x) = \sum_{i=1}^{n/4} \exp(x_{4i-1} - x_{4i-2})^2 + 100(x_{4i-2} - x_{4i-1})^6 + \tan(x_{4i-1} - x_{4i})^4 + x_{4i-3} + (x_{4i} - 1)^2
\]

Starting point: (1, 2, 2, 2,...............)^T

4. Cantrell function:
\[
f(x) = \sum_{i=1}^{n/4} \exp(x_{4i-1} - x_{4i-2})^4 + 100(x_{4i-2} - x_{4i-1})^6 + \tan^{-1}(x_{4i-1} - x_{4i})^4 + x_{4i-3}^8
\]

Starting point: (1, 2, 2, 2,...............)^T

5. Rosenbrock function:
\[
f(x) = \sum_{i=1}^{n/2} (100(x_{2i} - x_{2i-1})^2 + (1 - x_{2i-1})^2)
\]

Starting point: (-1.2,1,-1.2,1,......)^T

6. Beale function:
\[
f(x) = (1.5 - x_1(1 - x_2))^2 + (2.25 - x_1(1 - x_2)^2) + (2.652 - x_1(1 - x_2)^2)
\]

Starting point: (0,0, ............ .......)^T

7. Shallow function:
\[
f(x) = \sum_{i=1}^{n/2} ((x_{2i-1}^2 - x_{2i}x_{2i-1})^2 + (1 - x_{2i-1})^2)
\]

Starting point: (-2,............ ...........)^T

8. Wolfe function:
\[
f(x) = (-x_i(3 - x_i/2 + 2x_i - 1)^2 + \sum_{i=1}^{n/2} (x_{i-1} - x_i(3 - x_i(3 - x_i/2) + 2x_{i+1} - 1)^2 + (x_{i+1} - x_i(3 - x_i/2 - 1)^2
\]

Starting point: (-1, ............ .......)^T