J. KSIAM **Vol.6**, No.1, 91-107, 2002

SCALING METHODS FOR QUASI-NEWTON METHODS

ISSAM A.R. MOGHRABI

ABSTRACT. This paper presents two new self-scaling variable-metric algorithms. The first is based on a known two-parameter family of rank-two updating formulae, the second employs an initial scaling of the estimated inverse Hessian which modifies the first self-scaling algorithm. The algorithms are compared with similar published algorithms, notably those due to Oren, Shanno and Phua, Biggs and with BFGS (the best known quasi-Newton method). The best of these new and published algorithms are also modified to employ inexact line searches with marginal effect. The new algorithms are superior, especially as the problem dimension increases.

1. INTRODUCTION

The theoretical and practical merits of the quasi-Newton family of methods for unconstrained optimization have been systematically explored since the classic 1963 paper of Fletcher and Powell [1]. In the 1970's the self-scaling variable-metric algorithms were introduced, showing significant improvement in efficiency over earlier methods. In particular, in a series of papers, Oren [2,3], Oren and Luenberger [4], Oren and Spedicato [5], Shanno and Phua [6] developed these algorithms for minimizing an unconstrained nonlinear function f(x), where $x \in R^n$, $f \in R$ and the gradient g(x) is available for any given x. Variable-metric algorithms begin with an estimate x1 to the minimiser x_{min} and a numerical estimate H_1 of the inverse Hessian matrix, $G^{-1}(x)$. A sequence of points $\{x_k\}$ is then defined by:

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

where $g_k = \nabla f(x_k)$ and k is a scalar chosen so as to reduce the value of f at each iteration. H is updated by:

$$H_{k+1} = [H_k H_k y_k y_k^T H_k / y_k^T H_k y_k + \phi_k w_k w_k^T] + \rho_k s_k s_k^T / s_k^T y_k \tag{2}$$

Key Words: Unconstrained optimization, variable metric, self-scaling

with

$s_k = x_{k+1} - x_k$, $y_k = g_{k+1} - g_k$,

$$w_k = (y_k^T H_k y_k)^{1/2}[s_k/s_k^T y_k H_k y_k/y_k^T H_k y_k] \tag{3}$$

where ϕ_k and ρ_k are scalars.

The updating is performed so that:

$$H_{k+1} y_k = \rho_k s_k. \tag{4}$$

This condition is commonly satisfied with $\rho_k = 1$, and is then called "the Secant equation". With this restriction on (2) we have the so-called 'Broyden family' of algorithms [7]. For a quadratic function, G^{-1} is constant and satisfies $s_k = G^{-1}y_k$ for any corresponding y_k and s_k; clearly the objective of such updating formulae is that H_k tends (in some sense) to the inverse Hessian, $G^{-1}(x_k)$, for a general function. It is well-known that if f is a quadratic, and exact line searches are curried out, then after n iterations, $H_{n+1} = G^{-1}$. However, perhaps the strongest result concerning the convergence of the H-matrices towards G^{-1} for quadratic functions is that of Oren and Luenberger [4]. This is derived from the following result for variable metric methods using exact line search:

Theorem (I): If f, a quadratic objective function, is minimized by the sequence $\{x_k\}$ defined by the iteration $x_{k+1} = x_k - \lambda H_k g_k$, where x_1 is a given starting point and H_1 is any positive definite matrix, and where k minimizes $f(x_k - H_k g_k)$, then

$$f(x_{k+1}) - f(x_{min})[(K(R_k) - 1)/(K(R_k) + 1)]2(f(x_k) - f(x_{min})) \tag{5}$$

where

$R_k = G^{1/2} H_k G^{1/2}$, and K(.) denotes the condition number of any matrix. For proof see [4].

The quantity $[(K(R_k)-1)/(K(R_k)+1)]^2$, referred to by Oren and Luenberger as the "single-step convergence rate", is increased or decreased according as $K(R_k)$ is increased or decreased. Thus, the rate of convergence is maximized if $K(R_k)$ approaches 1.

Unfortunately the Broyden family cannot guarantee to reduce $K(R_k)$ at each iteration. Hence Oren and Luenberger introduced a preliminary step to the updating process whereby H_k is first scaled by a factor k. They are then able to prove:-

Property 1: Provided $k \in [0,1]$ and provided

$$\mu_k = \beta s_k^T H_k^{-1} s_k / s_k^T y_k + (l - \beta) s_k^T y_k / y_k^T H_k y_k \tag{6}$$

for some [0,1], then {K(R_k)} is monotonically decreasing and the eigenvalues of Rk tend monotonically to unity as k increases. For proof see [2].

lf an estimate of the inverse Hessian is maintained (rather than an estimate of the Hessian itself which is sometimes preferred) then there is a strong motivation for choosing β= 0 in (6). This gives:

$$\mu_k = s_k^T y_k / y_k^T H_k y_k \tag{7}$$

The most successful member of the Broyden family is the BFGS, which corresponds to a choice $\rho_k = 1$ in (2). We can now summarize the scaled BFGS algorithm due to Oren and Luenberger:-

Algorithm 1: (Oren)

Start with any initial point x_1.

Step 1: Set k=1 and choose H_1 to be any positive definite matrix (usually H_1 = I).

Step 2: Determine the step-size k to minimize f(x_k + λd_k) where d_k = - $H_k g_k$, and obtain x_{k+1} = x_k + λ_k d_k.

Step 3: Set

$$H_{k+1} = (H_k H_k y_k y_k^T H_k / y_k^T H_k y_k + w_k w_k^T)\mu_k + s_k s_k^T / s_k^T y_k \tag{8}$$

where w_k is defined in (3) and μ_k in (7).

Step 4: If not converged increase k by 1 and return to step 2.

Algorithm I possesses the following properties for a quadratic function:-

(a) If k minimizes f(x_k - $H_k$dk) for all k, then the vectors d_k are mutually conjugate (with respect to G) and hence the solution is obtained in at most n iterations.

(b) The condition number of the matrix R_k is strictly monotonically decreasing.

(c) lf k = 1 for all k, then the algorithm converges "two-step superlinearly", i.e.

$$limit_{k\to\infty} ||x_{k+1} - x_{min}|| / ||x_k - x_{min}|| = 0 \tag{9}$$

The proofs of these properties can be found in [8].
We also note that the value of k in (7) is chosen so that

$$\mu_k H_k y_k = s_k \tag{10}$$

Thus the Secant condition is satisfied before and after updating. The principal theoretical reservation to algorithm I is the failure in the quadratic case of the H-matrices to converge to G^{-1} in a finite number of steps. Experimental evidence in this paper supports the view of McCormick and Ritter [9] that the finite termination property for quadratics is desirable when minimizing general functions.

2. Shanno & Phua's Initial Self-Scaling Method

It would seem a desirable property that the sequence H_k (defined in (2)) should be invariant under scaling of $f(x)$ by a constant. This is indeed achieved by the self-scaling methods just described, but as Shanno and Phua [6] point out, it is only necessary to scale H at the first iteration to achieve this. They consider two possible scalings of H_1 and they show that in addition to providing the desired invariance, they also greatly enhance the numerical stability of members of Broyden's family.

In both of their scalings it is assumed that $H_1 = I$ may be used initially to determine x_2 where 1 is chosen according to some steplength or linear search criterion to ensure sufficient reduction in the function f. Once x_2 has been determined, but before H_2 is calculated, they scale H_1 to $H_1^+$and use H_1^+ instead of H_1 in the formula to define H_2. In their first alternative,

$$H_1^+ = \lambda_1 H_1 \qquad (\textbf{Algorithm I}) \tag{11}$$

and in their second alternative

$$H_1^+ = \mu_1 H_1 \qquad (\textbf{Algorithm II}) \tag{12}$$

with μ_1 given by (7). In either case, the Broyden family (BFGS in particular) becomes 'self-scaling' in this invariance sense without the need to apply k at subsequent iterations.Shanno and Phua's computational results show that for large problems the update (12) is preferable.

3. Biggs' Method

A consequence of the QN-condition is that

$$y_k^T H_{k+1} y_k = y_k^T s_k \tag{13}$$

Thus $y_k^T H_{k+1} y_k$ is matched to $y_k^T s_k$, a 'first-difference' estimate of the true directional curvature s_k^T $G(x_{k+l}) s_k$ of f at x_{k+l}. The estimate is exact if f is quadratic.
Biggs [10,11] observed that a more accurate estimate of this curvature can be obtained. Four independent pieces of information are available along s_k, namely the function values and directional derivatives at x_k and x_{k+l}. Thus a cubic model of f along s_k can be constructed, fielding an estimate $\rho_k y_k^T s_k$ of the curvature, where

$$\rho_k = s_k^T y_k / [4 s_k^T g_{k+1} + 2 s_k^T g_k - 6(f_{k+1} - f_k)] \quad (\textbf{Algorithm III}) \qquad (14)$$

Biggs therefore suggests the use of this value in the formula (2). Note that for a quadratic function, k = I and hence algorithm III is identical to the Broyden family.

4. A new self-scaling family of Methods

In this section we consider the Broyden family with $\phi_{k+1} = 1$, i.e. the BFGS update, and it will suffice to omit subscripts k, k+1 and simply use '*' to denote new values in the rest of this paper.
The standard BFGS update can be separated into two components, $H^{(1)}$ and $H^{(2)}$, so that $H^*_{BFGS} = H^{(1)} + H^{(2)}$, where

$$H^{(1)} = H - Hyy^T H / y^T Hy + ww^T \qquad (15)$$

$$H^{(2)} = ss^T / s^T y \qquad (16)$$

where w is the vector defined in (3), with subscripts omitted.
Oren and Biggs' modifications of the BFGS formula can then be written as

$$H^*_{Oren} = \mu H^{(1)} + H^{(2)} \qquad (17)$$

$$H^*_{Biggs} = H^{(1)} + \rho H^{(2)} \qquad (18)$$

This suggests a more general family of the form

$$H^* = \alpha H^{(1)} + \gamma H^{(2)} \qquad (19)$$

which will satisfy

$$H^{*}y = \gamma s \tag{20}$$

This relaxation of the QN-condition is of particular interest in deriving algorithms for non-quadratic objective functions.

Several choices of α and γ have been investigated but the most effective one (in numerical comparisons), presented here, is readily interpreted in terms of the earlier algorithms and their properties. We define

$$H^{*}_{new} = H^{(1)} + \sigma H^{(2)} \qquad \textbf{(Algorithm IV)} \tag{21}$$

where $\sigma = 1/\mu$.

The method is similar to Biggs in that only $H^{(2)}$ is scaled, but we now have

$$y^{T}H^{*}_{new}y = \sigma s^{T}y = y^{T}Hy \tag{22}$$

Unfortunately the nice property in Biggs' method of reverting to the Broyden family in the quadratic case is lost. However, it can also be viewed in terms of the Oren method as follows.

$$H^{*}_{new} = 1/\mu H^{*}_{Oren} \tag{23}$$

i.e. H is scaled by μ, the BFGS update is applied and then the resulting matrix is scaled by $1/\ \mu$.
Both BFGS and Oren's update generate identical conjugate gradient search directions provided that the function is *quadratic* and exact line searches are used. To prove that the new update IV will also satisfy this property, consider first:
Property 1 : Let f be given by

$$f(x) = \frac{1}{2}x^{T}Gx + b^{T}x, \tag{24}$$

where G is symmetric positive definite. Choose an initial approximation $H_1 = H$, where H is any symmetric positive definite matrix of appropriate order. Obtain H^{*}_{new} from H where d = - Hg is the search direction and assuming exact line searches then

$$H_{i+1}g^{*} = Hg^{*},\ for\ 0 \leq i < k \leq n \tag{25}$$

Proof: Apply induction on i. Let $H_1 = H$, on the above assumptions so that

$$H_1 g^* = Hg^* \tag{26}$$

Using the formula (IV) we have

$$H_{i+1}g^* = Hg^* y^T Hg^*/y^T Hy.Hy + w^T g^* w + \sigma s^T g^*/s^T ys \tag{27}$$

Now assuming that the property is true for i, namely, that

$$H_i g^* = Hg^* \tag{28}$$

we prove it is true for $i+1$ by using the following two standard properties (not proved here) which are known to be true for quadratic functions and exact line searches:

(a) $s_j^T g^* = 0$ for j=1,2,...,k

(b) $g_j^T H_j g^* = 0$ for j=1,2,...,k

(the orthogonality property satisfies when f is quadratic and exact line search is used). Substituting in (27) we have

$$H_{i+1}g^* = Hg^* \tag{29}$$

since $y_i^T H_i g^* = 0$ for $i < k$, and also

$$w_i^T g^* = (y_i^T H_i y_i)^{1/2}[s_i^T g^*/s_i^T y_i (H_i y_i)^T g^*/y_i^T H_i y_i = 0 \tag{30}$$

Since the property already holds for $i = 1$ the normal inductive proof is established for any i. We now prove the following theorem:

Theorem (II): Assume that f(x) be the quadratic function defined in (24) and that the line searches are exact: if H is any symmetric positive definite matrix (or appropriate order) and we define an updating

$$H^*_{new} = Hyy^T H/y^T Hy + ww^T + \sigma ss^T/y^T y \tag{31}$$

where $\sigma = y^T Hy / y^T s$, then the search direction

$$d^*_{new} = H^* g^* \tag{32}$$

is identical to the conjugate-gradient direction [12] d^*_{CG} defined by

$$d^*_{CG} = -g, \quad \text{for k=0 and } d^*_{CG} = [g^* + y^T g^*/y^T d]d, \text{ for k=1} \tag{33}$$

Proof: The update (31) can be written as:

$$H^*_{new} = H - sy^T H/s^T yHys^T/s^T y + (\sigma + y^T Hy/s^T y)ss^T/s^T y, \tag{34}$$

Now

$$d^*_{new} = Hg^* + (y^T Hg^*/s^T y)s + s^T g^*.Hy/s^T y - 2y^T Hy.sTg^*/(y^T s)^2 s \tag{35}$$

$$= -Hg^* + (y^T Hg^*/y^T s)s \tag{36}$$

using the property $s^T g^* = 0$, quoted earlier which holds for exact line searches.

The vector g^* can be substituted for Hg^* by using property 2. Therefore

$$d^*_{new} = g^* + (y^T g^*/y^T s_{new})s_{new} \tag{37}$$

We also know that d^*_{BFGS} and d^*_{CG} are identical (see Nazareth [13]), and d^*_{new} is identical to $d^*_{BFGS}$with exact line searches. Hence equation (38) becomes

$$d^*_{new} = g^* + (y^T g^*/y^T d^{CG})d^{CG} = d^*_{CG}, \tag{38}$$

and the proof follows.

5. A New Single Scaling : an Initial Scaling of H_1

Single scalings are normally applied at the start of the optimization procedure, and are based on the fact that choosing I (or other arbitrary matrix) as an estimate of G^{-1} is committing the algorithm to a sequence of poor estimates H of G^{-1}. We consider a simple initial scaling which is identical to the new scaling algorithm IV in the case of a quadratic function; for the general function comparative testing shows it can improve the performance of the new scaling algorithm IV.

We assume here that H_1 =I and that H_1 is used to determine λ_1 such that x_2= x_1 + $\lambda_1 d_1$ where λ_1 minimizes f(x_1-$\lambda_1 H_1 g_1$). Shanno's algorithm II applied to the initial step only can be written as

$$H^*_{Shanno} = \lambda H^{(1)} + H^{(2)} \tag{39}$$

where
$\lambda = l$, for k$\geq$ 2.
Multiplying equation (40) by the scalar, σ, yields
$H^*_{Snewh} = \lambda\sigma\, H^{(1)} + \sigma H^{(2)}$ **(Algorithm V)**
where
$\lambda\ \sigma$= 1 for k$\geq$ 2.
Theorem (III): Let f be a quadratic function defined as in (24). Assume that H_1 = I and exact line search is used; then we have $\lambda_1\ \sigma_1$ = 1 and the algorithms IV and V are identical.
Proof:
For the QN-method

$$d_1 = -H_1 g_1 \Longrightarrow s_1 = -\lambda_1 g_1. \tag{40}$$

Since $H_1 = I$, it follows that $s_1^T\ s_1$ = - $\lambda_1 g_1^T\ s_1$.
Now
$\sigma_1 = y_1^T H_1 y_1 / y_1^T s_1 = y_1^T y_1 / y_1^T s_1$.
since H_1 =I,

$$\lambda_1\sigma_1 = (s_1^T s_1/g_1^T s_1)(y_1^T y_1/y_1^T s_1). \tag{41}$$

Now substitute (41) in (42) to get:
$\lambda_1\ \sigma_1 = (g_1^T g_1/g_1^T g_1)(y_1^T y_1/y_1^T g_1) = y_1^T y_1/y_1^T y_1$= 1.0
(since exact line search is used).

6. A Second Self-Scaling algorithm using Non-Exact Searches

In this section the Brodlie analysis [14], which he applied to the Broyden family (including BFGS), is applied directly to the self-scaling BFGS algorithms, with the one-parameter family of correction formulae. We define the updating

$$H^* = (H - Hyy^T H/y^T Hy + ww^T) + ss^T/y^T y \tag{42}$$

where w is a vector defined in (3), and y, H, s as defined earlier.
Proceeding exactly as in [14] we can show that the new search direction $d^* = -H^*g^*$ for such algorithms can be expressed as a combination of two vectors p_1 and p_2. Now the vector H^*g^* can be written as:

$$H^*g^* = [Hg^*y^THg^*/y^THy.Hy{+}(y^THy)\{s^Tg^*/s^Tyy^THg^*/y^THy\}$$
$$\{s/s^Ty - Hy/y^THy\}]\theta + \rho(s^Tg^*/s^Ty)s \quad (43)$$

If we let

$$[s^Tg^*/s^Ty] - y^THg^*/y^THy = \tau \quad (44)$$

and express

$$Hy = Hg^* - Hg = Hg^* + s/\lambda \quad (45)$$

then the expression (44) gives:

$$H^*g^* = [Hg^* - (y^THg^*/y^THy).Hg^* - \tau Hg^* - y^THg^*/(\lambda y^THy)s$$
$$+\tau(y^THy/y^Ts - 1/\lambda)s] + (s^Tg^*/s^Ty)s \quad (46)$$

But again using (46) we can show that

$$[y^THy/y^Ts]1/\lambda = (\lambda y^THy - s^Ty)/\lambda s^Ty = y^THg^*/s^Ty \quad (47)$$

so that, by substitution,

$$H^*g^* = [(1 - y^THg^*/y^THy - \tau)\vartheta)]Hg^*+$$
$$(\rho s^Tg^*/s^Ty - y^THg^*/\lambda y^THy\vartheta + \tau\vartheta y^THg^*/s^Ty)s. \quad (48)$$

Again, substituting for $1/\lambda$ we have

$$y^THg^*/\lambda y^THy = -(y^THg^*/y^THy)(y^THg/s^Ty) = [y^THg^*/s^Ty](1 - y^THg^*/y^THy) \quad (49)$$

Hence (49) becomes:

$$H^*g^* = \theta(1 - y^T Hg^*/y^T Hy - \tau)[Hg^* - (y^T Hg^*/s^T y)s] + \rho(s^T g^*/s^T y)s \tag{50}$$

Let $\psi = \theta(1 - y^T Hg^*/y^T Hy - \tau)$
then

$$H^*g^* = \psi[Hg^* - (y^T Hg^*/s^T y)s] + \rho(s^T g^*/s^T y)s$$

$$= \psi[1 - sy^T/s^T y]Hg^* + (\rho s^T g^*/s^T y)s \tag{51}$$

Now

$$d^* = -H^*g^* = -\rho p_1 - \psi p_2 \tag{52}$$

where

$$p_1 = (s^T g^*/s^T y)s, \text{and } p_2 = [I - sy^T/s^T y]Hg^*. \tag{53}$$

Equation (54) therefore gives a family of self-scaled BFGS algorithms.
In particular we concentrate on the specific new self-scaling algorithm defined by

$$\rho = \sigma \text{ and } \theta = 1, (\text{where } \sigma \text{ is defined earlier}) \ (\textbf{Algorithm VI}). \tag{54}$$

However, Oren's algorithm, modified to incorporate inexact searches, is defined by

$$\rho = 1 \text{ and } \theta = \mu, \qquad (\textbf{AlgorithmVII}) \tag{55}$$

where is the scalar defined in (7).

Again, for Shanno and Phua's algorithm we have

$$\rho = 1 \text{ and } \theta = \mu_1 \text{ or } \lambda_1, \text{ for } k = 1 \text{ and } \rho = 1, \ \theta = 1, \text{ for } k \geq 2. \tag{56}$$

and finally for our new initial self-scaling algorithm

$$\rho = \sigma \text{ and } \theta = \lambda_1\sigma_1 \text{ (for k=1)} \tag{57}$$

and all set to one for k$\geq$ 2.
In (54) with exact line searches, $s^T g^* = 0$, and hence we have p_1 null. In general, however, p_1 is not null and the quantity p_1 may be regarded as a correction term for the inexact search. From the family of self-scaled BFGS algorithms we select those defined by equations (56) and (57) for our comparative numerical tests.

7. Computational Results

The comparative tests involve twenty six well-known test functions (given explicitly in the Appendix). The comparative performances of the algorithms are evaluated by considering both the total number of iterations and the number of function evaluations. We define 'iteration' to mean the step carrying a point x along the direction $d = -Hg$ to a new point $x*$, and the number of function calls quoted is that required to reduce the value of f(x) below 1.0^{-10}. The cubic interpolation technique, fully described in [15], is used as the linear search subprogram unless stated otherwise.

Five algorithms are tested and compared: (i) the original BFGS algorithm (ii) Oren's algorithm I (iii) Shanno's algorithm II with 1 as an initial sealing factor (iv) algorithm IV and (v) algorithm V were compared over the twenty six test functions. For convenience, our numerical results for exact line searches are presented in two tables: *table1* contains the results for dimensionality $n < 10$, while *table 2* contains the results for problems in the range 10$\leq$n$\leq$100.

For the first group (*table 1*) all the established algorithms perform about the same except for poor results with Shanno's method on Oren and Spedicato's power function (already reported by Shanno [6]). The new self-scaling algorithms save about 30% in NOF with negligible additional overheads.
For the second group (*table 2*) all the self-scaling methods perform much better than BFGS, but the new algorithms again save about 30% in NOF on the best established self scaling algorithm.

For inexact line searches our numerical results, presented in *table 3*, show that the new algorithm VI beats the modified Oren's algorithm VII by about 30% in the number of function calls although the two algorithms perform about the same when compared by the total number of iterations. For completeness, the best new self-scaling algorithm in its exact line search form (IV) is compared with the latest NAG self-scaling algorithm OPVM (Biggs' algorithm), released March 1989, with numerical results in Table 4. Since the line searches are different a measure of total work is included in the comparison: on this measure the new algorithm serves about 67% on TWORK overall, but this saving is entirely at higher dimensions since Biggs'is more efficient at low dimensionality. The theoretical observation, e.g. in [6] Shanno and Phua that for some extended test functions (like Rosenbrock's or Powell's) the number of iterations and function evaluations must remain constant with increasing n with perfect arithmetic confirmed in our numerical computation.

Finally, the computational results presented here show that while self-scaling generally improves computational efficiency on small problems (say within < 4) it can sometimes decrease efficiency : however, it normally improves the performance of the BFGS method on large problems, and the relative improvement increases monotonically with dimensionality n. The effect of changing to inexact line searches is marginal. Oren's algorithm requires slightly more NOF, whereas the new algorithm IV slightly less. There are no significant variations with particular test functions. In conclusion, for this particular set of test functions and at the required accuracy levels, the new algorithms perform very much better than the several well-known methods.

All algorithms terminate when | f - f_{min} | < 1 x 10^{-10}

TABLE : (1)

		BFGS	OREN	SHANNO	NEWH	SNEWH
TEST Function	N		ALG(I)	ALG(II)	ALG(IV)	ALG(V)
		NOI(NOF)	NOI(NOF)	NOI(NOF)	NOI(NOF)	NOI(NOF)
ROSEN	2	22(72)	21(89)	21(72)	21(59)	21(59)
CUBE	2	15(60)	22(85)	22(77)	24(65)	21(59)
BEALE	2	9(26)	9(33)	9(26)	8(20)	8(20)
BOX	2	8(38)	8(43)	8(39)	8(43)	8(44)
FREUD	2	6(21)	7(34)	6(25)	6(18)	6(18)
RECIPE	3	5(17)	6(21)	5(18)	6(19)	6(19)
PIGS	3	11(39)	13(47)	10(36)	11(31)	10(27)
HPLICAL	3	19(59)	19(72)	19(61)	18(41)	21(49)
POWELL	4	18(76)	29(99)	26(110)	30(66)	30(66)
WOOD	4	57(162)	18(66)	19(73)	18(42)	17(40)
MIELE	4	29(112)	30(103)	29(120)	31(100)	30(93)
TIXON	10	29(73)	18(56)	17(46)	18(40)	18(40)
OREN	10	23(122)	12(55)	48(294)	12(64)	12(64)
TOTAL	NOI	253	212	239	211	208
	(NOF)	(877)	(803)	(997)	(608)	(598)

TABLE : (2)

TEST FUNCTION	N	BFGS	OREN ALG.(I)	SHANNO ALG.(II)	NEWH ALG.(IV)	SNEWH ALG.(V)
		NOI(NOF)	NOI(NOF)	NOI(NOF)	NOI(NOF)	NOI(NOF)
NON-DIGN	20	40(102)	22(91)	20(73)	20(61)	20(61)
OREN	30	71(374)	21(85)	125(747)	21(95)	21(93)
TRI-DIGN	30	28(57)	28(85)	28(85)	28(57)	28(57)
FULL	40	39(79)	39(123)	39(123)	39(79)	39(79)
SHALLOW	40	8(25)	8(30)	8(30)	8(18)	8(18)
EX-POWEL	60	32(101)	27(115)	27(115)	36(79)	35(77)
EX-WOOD	60	167(452)	19(73)	19(73)	18(42)	17(40)
EX-ROSEN	60	169(485)	22(75)	22(75)	22(61)	22(61)
WOLFE	80	63(127)	37(114)	37(114)	37(75)	37(75)
EX-POWELL	80	35(105)	27(115)	27(115)	39(85)	39(85)
NON-DIGN	90	46(112)	20(74)	20(74)	21(61)	21(61)
EX-WOOD	100	230(641)	20(73)	19(73)	18(42)	17(40)
EX-ROSEN	100	249(712)	22(75)	22(75)	22(61)	22(61)
TOTAL	NOI	1177	330	413	329	326
	(NOF)	(3372)	(1772)	(1772)	(816)	(808)

TABLE : (3)

TEST		ALGORITHM(6)	ALGORITHM(7)
FUNCTIONS	N	NOI(NOF)	NOI(NOF)
ROSEN	2	22(54)	25(97)
CUBE	2	22(57)	24(92)
BEALE	2	8(20)	9(33)
BOX	2	9(41)	9(44)
FREUD	2	6(18)	7(34)
BIGGS	3	11(31)	13(47)
HELICAL	3	18(39)	18(73)
RECIPE	3	6(19)	6(21)
MIELE	4	30(83)	30(103)
POWILL	4	31(67)	29(99)
WOOD	4	19(42)	18(66)
DIXON	10	17(37)	18(56)
OREN	10	12(64)	12(55)
NON-DIGN	20	20(46)	22(84)
TRI-DIGN	30	28(57)	28(83)
OREN	30	21(95)	21(85)
SHALLOW	40	6(18)	6(30)
FULL	40	39(79)	39(121)
EX-ROSEN	60	23(57)	26(101)
EX-POWELL	60	40(83)	35(119)
EX-WOOD	60	60(42)	18(66)
EX-POWELL	80	43(88)	39(139)
WOLFE	80	80(75)	37(112)
NON-DIGN	90	23(53)	22(95)
EX-WOOD	100	19(42)	18(66)
EX-ROSEN	100	23(57)	26(101)
TOTAL	NOI	551	555
	(NOF)	(1364)	(2022)

References

[1] R. Fletcher and M.J.D. Powell, "A rapidly convergent descent method for minimization", Computer Journal 6 (1963), 163-168.

[2] S.S. Oren, "Self-scaling variable metric algorithm, Part II", Management Science 20 (1974), 863-874.
[3] S.S. Oren, "On the selection of parameters in self-scaling variable metric algorithms", Mathematical Programming 3 (1974) 351-367.
[4] S.S. Oren and D.G. Luenberger, "Self-scaling variable metric algorithm, Part I", Management Science 20 (1974) 845-862.
[5] S.S. Oren and E. Spedicato, "Optimal conditioning of self-scaling variable metric algorithms", Mathematical Programming l0 (1976) 70-90.
[6] D.F. Shnnno and K.H. Phua, "Matrix conditioning and nonlinear optimization", Mathematical Programming14 (1978) 149-160.
[7] C.G. Broyden, "The convergence of a class of double rank minimization algorithms II. The new algorithm", Journal of the Institute of Mathematics and its Applications 6 (1970) 221-231.
[8] S.S. Oren, "Self-scaling variable metric algorithm without line search for unconstrained minimization", Mathematics of Computation 27 (1973), 873-885.
[9] G.P. McCormick and K.. Ritter, "Methods of conjugate directions versus quasi-Newton methods", Mathematical Programming 3 (1972) 101-l16.
[10] M.C. Biggs, "Minimization algorithms making use of non-quadratic properties of the objective function", Journal of the Institute of Mathematics and its Applications 8 (1971) 315-327.
[11] M.C. Biggs, "A note on minimization algorithms which make use of non-quadratic properties of the objective function", Journal of Institute of Mathematics and its Applications12 (1973) 337-338.
[12] H.R. Hestenes and E. Stiefel, "Methods of conjugate gradients for solving linear systems", Journal of Research of the National Bureau of Standards, 49 (1952), 409-436.
[13] L. Nazareth, "A relationship between BFGS and conjugate-gradient algorithms and its implementations for new algorithms", SIAM Journal on Numerical Analysis,16 (1979), 794-800.
[14] K.W. Brodlie, "Some topics in unconstrained minimization", Ph.D. thesis, University of Dundee, (1973).
[15] B. Bunday, "A Basic Optimization Methods", Edward Arnold, Bedford Square, London, (1984).

Appendix

All the test functions used in this paper are from general literature

1. Rosenbrock banana function, n=2,
$f=100\ (x_2-x_1^2)^2+(1-x_1)^2$, $x_0=(-1.2,1.0)^T$.
2. Cube function, n=2,
$f=100\ (x_2-x_1^3)^2+(1-x_1)^2$, $x_0=(-1.2,1.0)^T$.
3. Scale function, n=2,
$f=(1.5-x_1\ (1-x_2))^2+\ (2.25 - x_1\ (1 - x_2^2))^2+\ (2.625 - x_1(1 - x_2^3))^2$, $x_0 = (0,0)^T$.
4. Box function, n = 2,
$f = \sum_{i=1}^{n}(\ e^{-x1}z_i$ - e -$x^2 z_i$ - $e^{-zi} + e\text{-}^{10zi})\ ^2$,where $z_i = (0.1)^i$ and $x_0 = (5,0)^T$, i=1,...
5. Frudenstein and Roth function, n=2,
$f = [-13 + x_1 + ((5-x_2)x_2-2)x_2]^2 + [-29 + x_1 + ((1+x_2)x_2-14)x_2]^2$, $x_0 = (30,3)^T$.
6. Recipe function, n = 3,
$f = (x_1-5)^2 + x_2^2 + x_3^2/(x_1-x_2)^2$, $x_0 = (2,5,1)^T$.
7. Biggs function, n=3,
$f = \sum_{i=1}^{n}(e^{-x1zi} - x_3\ e^{-x2zi} - e^{-zi} + 5e^{-10zi})\ ^2$, where $z_i = (0.1)$ i and $x_0= (1,2,1)^T$,i=1...
8. Helical Valley function, n=3,
$f = 100\ \{\ [x_3-1.0\ ]^2 + [r-1]^2\ \}+ x_3^2$,where r=1/2 arctan $(x_2/x_l\ )$, for $x_l > 0$

and r = 1/2 + 1/2 arctan (x_2/x_l) for $x_l < 0$, $x_0 = (-1,0,0)^T$.
9. Miele and Cornwell function, n = 4,
$f = (e^{x_1} - 1)^2 + \tan 4 (x_3 - x_4) + 100 (x_2- x_3)^2 + 8x_1+ (x_4- l)^2$, $x_0= (1, 2, 2, 2)^T$.
10. Dixon function, n = 10,
$f = (1- x_l)^2 + (1- x_{10})^2 + \sum_{i=1}^{n}(x_i^2 - x_i+1)^2$, $x_0 = (-1;..)^T$, i=2,...
11. Oren and Spedicato power function, n = 10,30,
$f = \sum_{i=1}^{n}(i - x_i^2)^2$, $x_0 = (1,.....)^T$.
12. Non diagonal variant of Rosenbrock function, n = 20, 90,
$f = \sum_{i=1}^{n}[100 (x_l - x_i^2)^2 + (1 - x_i)^2]$, $x_0 = (-1, ..)^T$, i=1,..
13. Tri-diagonal function, n = 30,
$f = [\sum_{i=2}^{n}(2x_i - x_{i-1})^2]$, $x_0 = (1;....)^T$.
14. Full set of distinct eigenvalues problem, n = 40,
$f = (x_l-1)^2 + \sum_{i=2}^{n}(2x_i - xi_{-1})^2$, $x_0 = (1;..)^T$.
15. Shallow function (Generalized form), n = 40,
$f = \sum_{i=1}^{n/2}(x_{2i-1}^2 - x_{2i})^2 + (1 - x_{2i-1})^2$, $x_0 = (-2;.)^T$.
16. Powell function (Generalized form), n = 60, 80,
$f = \sum_{i=1}^{n/4}[(x_{4i-3} + l0\, x_{4i-2})^2 + 5 (x_{4i-1} - x_{4i})^2 + (x_{4i-2}-2 x_{4i-1})^4 + 10 (x_{4i-3} - x^{4i})^4]$,
$x_0 = (3,-1,0,1;...)^T$.
17. Wood function(Generalized form), n = 60,100,
$\sum_{i=1}^{n/4}f = [100 (x_{4i-2} - x_{4i-3}^2)^2 + (1- x_{4i-3})^2+ 90 (x_{4i} - x_{4i-1}^2)^2 + (1 - x_{4i-1})^2$
$+ 10.1 (x_{4i-2} - 1)^2 + (x_{4i}- 1)^2 + 19.8 (x_{4i-2} - 1)(x_{4i-l})$, $x_0 = (-3,-1;-3,-1,...)^T$.

Math. Dept., Faculty of Science
Beirut Arab University
P.O. Box 11-5020, Beirut, Lebanon
Email: imoghrabi@bau.edu.lb