International Journal of Applied Mathematics

Volume 26 No. 4 2013, 411-424
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)
doi: http://dx.doi.org/10.12732/ijam.v26i4.1

A NEW WEIGHTED CONJUGATE GRADIENT METHOD

Issam A.R. Moghrabi
MBA Office/M.I.S. Department
College of Business Administration
Gulf University for Science and Technology (GUST)
P.O. Box 7207, Hawally 32093, KUWAIT

Abstract: A new weighted multi-update conjugate gradient method is proposed for unconstrained optimization which is considered as memoryless variable metric methods. It is derived for inexact line searches and evaluated numerically against Shanno's two memoryless quasi-Newton methods. The numerical results indicate that, in general, the new method is numerically superior to Shanno's methods.

AMS Subject Classification: 65K10
Key Words: unconstrained optimization, conjugate gradient methods, variable metric methods, multiple update methods

1. Introduction

The Conjugate Gradient (CG) methods were first used to solve the general unconstrained minimization problem by Fletcher and Reeves [8] in 1964. They are still preferred to the more rapidly convergent Quasi-Newton methods (QN) for problems of high dimensionality since they only require storage of few vectors of length n unlike the $O(n^2)$ QN methods. This is one justification for further improving on the existing CG methods. We seek greater efficiency here by modifying Shanno's algorithms, at the expense of storing two $O(n)$ vectors.

Received: July 25, 2012

© 2013 Academic Publications

For a given symmetric positive definite matrix A, the finite set of non-null vectors $d_1, d_2, d_3, \ldots d_k$ is said to be conjugate if

$$d_i^T A d_k, \quad \text{for all} \quad i \neq k. \tag{1}$$

It is well known that such vectors are linearly independent.

The CG-methods are iterative and generate a sequence of approximation to the minimum x_{min} of a scalar function $f(x)$ of the vector variable x. The sequence x_k is defined by

$$x_{k+1} = x_k + \lambda_k d_k \tag{2}$$

and

$$d_{k+1} = -g_{k+1} + \beta_k d_k, \tag{3}$$

where g_k is the gradient of $f(x)$, λ_k is a positive scalar chosen to minimize $f(x)$ along the search direction d_k, and α_k is defined by

$$\beta_k = (y_k^T g_{K+1)}/(y_k^T d_k), \tag{4}$$

where $y_k = g_{k+1} - g_k$, and the definition of α_k in (4) is that due to Hestenes and Stiefel [9].

Perry [15] derived a CG algorithm as follows: he noted that in (3) the scalar β_k was chosen to make d_k and d_{k+1} conjugate using an exact line search (ELS). Since, in general, line searches are not exact, Perry rewrote (3) in view of this assumption as

$$d_{k+1} = -[1 - (d_k y_k^T)/(y_k^T d_k)]g_{k+1}. \tag{5}$$

The projection matrix multiplying g_{k+1} is not of full rank; hence, (5) is modified to

$$d_{k+1} = -[1 - (v_k y_k^T)/(y_k^T v_k) + (v_k v_k^T)/(v_k^T y_k)]g_{k+1} = Q_{k+1} g_{k+1}, \tag{6}$$

where $v_k = x_{k+1} - x_k$.

Perry gave other reasons to support his choice of the new term in (6). First, the matrix Q_{k+1} satisfies a relation similar to, but not identical with, the quasi-Newton (QN) condition, namely

$$Q_{k+1}^T y_{k+1} = v_k. \tag{7}$$

Also, (6) reduces to (5) if an ELS is carried out on that iteration.

Perry's limited experiments with his algorithm (six test functions with $n \leq 4$) showed that it performs only slightly better than the standard CG-method.

2. Memoryless Quasi-Newton Methods in CG Methods

2.1. Shanno's Method

Shanno [17] addressed the issue that (6) does not satisfy the actual QN-condition, which requires an update to the approximation of the inverse Hessian H_{k+1} in such a way as to satisfy

$$H_{k+1}^T y_k = v_k. \tag{8}$$

He also pointed out that the matrix Q_{k+1} is not necessarily symmetric or positive definite so that (6) may not define a downhill direction. Hence, he symmetrized $Q_{k+1}$by adding an appropriate term. Specifically, Shanno proposed:

$$Q_{k+1}^+ = [1 - (v_k y_k^T)/(y_k^T v_k) - (y_k v_k^T)/(y_k^T v_k) + (v_k v_k^T)/(y_k^T v_k)]. \tag{9}$$

But this new symmetric matrix satisfies neither (7) nor (8), so again a modified version was derived as follows

$$Q_{k+1} = 1 - [(v_k y_k^T + y_k v_k^T)/v_k^T y_k] + [1 + (y_k^T y_k/v_k^T y_k)][v_k v_k^T/(v_k^T y_k)]. \tag{10}$$

This form of the projection matrix Q_{k+1} has a special relationship with the BFGS updated formula

$$H_{k+1} = H_k - (H_k y_k v_k^T + v_k y_k^T H_k)/v_k^T y_k + [1 + y_k^T H_k y_k/v_k^T y_k][v_k v_k^T/v_k^T y_k]. \tag{11}$$

It is then easily seen that (10) is equivalent to (11) with H_k replaced with I. In fact, a similar dual relationship to (10)-(11) can be exhibited for any member of Broyden's θ-class update, [5].

The CG-method, which is referred to as a memoryless BFGS method defined by (10), namely

$$d_{k+1} = -Q'_{k+1} g_{k+1} \tag{12}$$

reduces again to (5) assuming ELS. Moreover, it does not require storage of the matrix $Q\prime_{k+1}$ since

$$\begin{aligned} d_{k+1} &= -g_{k+1} - [(1 + y_k^T y_k/v_k^T y_k)(v_k^T g_{k+1}/v_k^T y_k) \\ &\quad -(y_k^T g_{k+1}/v_k^T y_k)]v_k + (v_k^T y_k)y_k. \end{aligned} \tag{13}$$

Hence, no additional information is needed to compute d_{k+1} beyond that required by the standard CG-method.

2.2. Shanno's Self-Scaling Conjugate Gradient Method

The idea of self-scaling was originally developed in a series of papers by Oren [12], [13] and Oren and Spedicato [14]. Considering Broyden's θ-class of algorithms, Oren established in [8] the modification of the Broyden update formula to give

$$H_{k+1} = [H_k - (H_k y_k y_k^T H_k)/(y_k^T H_k y_k) + \theta_k \omega_k \omega_k^T]\eta_k + (v_k v_k^T)/(v_{kT} y_k), \quad (14)$$

where

$$w_k = (y_k^T H_k y_k)^{1/2}[v_k/v_k^T y_k - H_k y_k/y_k^T H_k y_k, \quad (15)$$

and θ_k is a free parameter; for BFGS, $\theta_k = 1$ and η_k is the scaling factor defined by

$$\eta_k = (v_k^T y_k)/(y_k^T H_k y_k) \quad (16)$$

as quoted in [14].

However, it seems natural to scale the CG-method on every iteration by using relation (13): this amounts to substituting I for H_k in (14). Thus, Shanno defined a modified CG-method with H_k replaced by I and

$$\begin{aligned} d_{k+1} \quad = \quad & -(v_k^T y_k/y_k^T y_k)g_{k+1} \quad (17)\\ & -(2v_k^T g_{k+1}/v_k^T y_k - y_k^T g_{k+1}/y_k^T y_k)v_k + (v_k^T g_{k+1}/y_k^T y_k)y_k. \end{aligned}$$

This corresponds to scaling the memoryless BFGS (10) and (12) by η_k.

Unfortunately, Shanno found in [15] that this modified CG-method did not produce good results as the one defined by (13).

2.3. A Conjugate Gradient Method As A Memoryless Variable Metric Update

A new family of variable metric (VM) methods can be obtained by multiplying Oren's update (14) at each iteration by a scalar $\sigma_k > 0$ such that

$$\sigma_k = 1/\eta_k, \quad (18)$$

which yields

$$H_{k+1} = H + [(2y_k^T H_k y_k)/y_k^T v_k)^2 v_k v_k^T [H_k y_k v_k^T + v_k y_k^T H_k]/y_k^T v_k. \quad (19)$$

The parameter σ_k is added to make the sequence invariant to scaling the objective function by a constant. Thus, it is reasonable to ensure scaling of the sequence by substituting σ_k, defined in (18), into (19). Using I for H_k in the right hand side of (19) we get the following new memoryless VM-method

$$d_{k+1} = -H_{k+1} g_{k+1} \tag{20}$$

or, by substitution for H_{k+1}:

$$\begin{aligned} d_{k+1} \quad &= \quad -g_{k+1} - [2y_k^T y_k v_k^T g_{k+1})/(y_k^T v_k)^2 (y_k^T g_{k+1})/(y_k^T v_k)] v_k \\ &\quad +(v_k^T g_{k+1})(v_k^T y_k)] y_k. \end{aligned} \tag{21}$$

We also note that if $v_k T g_{k+1} = 0$ (for E.L.S), then (21) reduces to

$$d_{k+1} = -g_{k+1} + (y_k^T g_{k+1})/(v_k^T y_k) d_k \tag{22}$$

which is the standard Hestenes and Stiefel CG-method [12] and therefore has n-step convergence to the minimum of a quadratic function. Thus the CG-method is defined precisely by the new VM update (19), where the approximation to the inverse Hessian is reset to the identity matrix at every step.

3. Double Update Memory Variable Metric Methods

3.1. Beal's Retart Critrion and Shanno's Double Update Memoryless Quasi-Newton Algorithms

Analysis of the rate of convergence for the CG-algorithm defined by (2)-(4) demonstrated that these algorithms generally exhibit a linear rate of convergence unless restarted (generally every n steps) with direction $d_t = -g_t$, (see [12]). As the step in the direction of the negative gradient frequently produces a very small reduction in the objective function, Beale [1] derived a restart criterion which has frequently improved convergence rate properties, but allows the restart step to use the computed direction d_t rather than restarting with $-g_t$ subsequent non-restart steps are defined by:

$$d_{k+1} = -g_{k+1} + \beta_k d_k + \gamma_k d_t, \tag{23}$$

where

$$\beta_k = (y_k^T g_{k+1})/(d_k^T y_k) \tag{24}$$

and

$$\gamma_k = (y_t^T g_{k+1})/(d_t^T y_t), \tag{25}$$

for $k = t+1, t+2, \ldots, t+n-1$.

However, in [10] this restart procedure was triedwith disappointing numerical results. Powell [16] suggests a modification to the Beal's restart procedure with the update (23) every n steps or whenever

$$\mid g_{k+1}^T g_k \mid \geq 0.2 \mid g_{k+1}^T g_{k+1} \mid . \tag{26}$$

Since successive gradients are necessarily orthogonal in Beale's method with ELS when$f(x)$ is quadratic. The condition (26) restricts convergence towards a point with a non-zero gradient.

Restarting may also be appropriate when the direction (23) is not sufficiently downhill. Powell, therefore, recommends restarting whenever

$$-1.0 \mid g_{k+1}^T g_k \mid \geq d_{k+1}^T g_{k+1} \geq -0.8 \mid g_{k+1}^T g_{k+1} \mid . \tag{27}$$

Computational results in [14] show that Beale's method with Powell's restarting criterion is consistently, though not markedly, superior to the CG-method with Fletcher's restarting technique (i.e., restarting with $-g_k$ every n steps).

Following the good published performance of the Powell restart procedure [last ref.], Shanno then examined the three-term recurrence (23) on which it is based: this took particular account of the dual CG-VM relationship studied in (10) and (11). Thus, (23) is rewritten as:

$$d_{k+1} = -[I(d_k y_k^T / d_k^T y_k)(d_t y_k^T / d_t^T y_t)] g_{k+1} - P_{k+1} g_{k+1}, \tag{28}$$

where t is the index of the last restart and the matrix P_{k+1} uses information from two prior points x_k and x_t in Beal's method we recall that the information gathered at x_t is critical and must be retained. At the same time, the storage requirement for the CG-algorithm must stay within order n locations. Hence, Shanno defined, for $k > t$

$$H_t' = [I \frac{v_t y_t^T + y_t v_t^T}{v_t^T y_t} + (1 + \frac{y_t^T y_t}{v_t^T y_t}) \frac{v_t^T v_t}{v_t^T y_t} \tag{29}$$

and

$$H_{k+1}' = H_t' - \frac{v_k y_k^T H_t' + H_t' y_k v_k^T}{v_k^T y_k} + (1 + \frac{y_k^T H_t' y_k}{v_k^T y_k}) \frac{v_k v_k^T}{v_k^T y_k}. \tag{30}$$

The search direction at x_{k+1} is then found by setting

$$\begin{aligned} d_{k+1} &= -H_{k+1}' g_{k+1} \\ &= -H_t' g_{k+1} + \frac{v_k^T g_{k+1}}{v_k^T y_k} H_t' y_k \\ &\quad -((1 + \frac{y_k^T H_t' y_k}{v_k^T y_k}) \frac{v_k^T g_{k+1}}{v_k^T y_k} - \frac{y_k^T H_t' g_{k+1}}{v_k^T y_k}) v_k. \end{aligned} \tag{31}$$

The vector $H\prime_t g_{k+1}$ and $H\prime_t y_k$ are defined by

$$H'_t g_{k+1} = g_{k+1} - \frac{v_t^T g_{k+1}}{v_t^T y_t} y_k + ((1 + \frac{y_t^T y_t}{v_t^T y_t}) \frac{v_t^T g_{k+1}}{v_t^T y_t} - \frac{y_t^T g_{k+1}}{v_t^T y_t}) v_t \tag{32}$$

and

$$\begin{aligned} H'_t y_t &= y_t - \frac{v_t^T y_k}{v_t^T y_t} y_t \\ &+ \left((1 + \frac{v_t^T y_t}{v_t^T y_t}) \frac{v_t^T y_k}{v_t^T y_t} - \frac{y_t^T y_k}{v_t^T y_t} \right) v_t. \end{aligned} \tag{33}$$

In implementing this algorithm additional storage is required to store vectors (32) and (33). The required vectors are thus x_{k+1}, x_k, g_{k+1}, g_k,d_k,d_t, and y_t (a total storage still of order n) while (31)-(33) is a direct two-vector analogue of (10).

Another far more successful search direction, proposed by Shanno, is generated by using Oren's update (14) in (28), yields for $k > t$

$$H'_t = [I - \frac{v_t y_t T \_ y_t v_t T}{v_t^T y_t} + \frac{y_t y_t^T}{v_t^T y_t} * \frac{v_t v_t^T}{v_t^T y_t}] \eta_t + \frac{v_t v_t^T}{v_t^T y_t}. \tag{34}$$

Shanno therefore suggested scaling the matrix $H\prime_t$ with

$$\eta_t = (v_k^T y_k)/(y_t^T y_t) \tag{35}$$

but not the matrix H_{k+1}. In this case the two additional vectors are defined by

$$H'_t g_{k+1} = \eta_t g_{k+1} - \frac{v_t^T g_{k+1}}{y_t^T y_t} y_t + (\frac{2 v_t^T g_{k+1}}{v_t^T y_t} - \frac{v_t^T g_{k+1}}{y_t^T y_t}) \tag{36}$$

and

$$H'_t y_k = \eta_t y_k - \frac{v_t^T y_k}{y_t^T y_t} y_t + (\frac{2 v_t^T y_k}{v_t^T y_t} - \frac{y_t^T y_k}{y_t^T y_t}). \tag{37}$$

However, he also tested the application of the Flecher [8] scaling in his numerical trials. Consequently, he proposed using only the scaled $H\prime_t$ at restart steps, and at each non-restart step to scale according to the following Flecher scaling criterion:

$$d'_{k+1} = [2(f_{k+1} - f_k)/d_{k+1}^T g_{k+1}] d_{k+1}. \tag{38}$$

Finally, we outline how Shanno implemented his algorithms.

Given x_1, set

$$d_1 = -g_1/(g_1^T g_1) \tag{39}$$

and for $k = 2$, the algorithms are restarted according to Powell's restarting criterion (26) or, by default, every n iterations. For the double update BFGS algorithm the restarting directions are defined by (13): the other steps use (31) and (33) and they are scaled by (38). However, for the scaled double update algorithm the restarting direction are defined by (17). The other steps use (31),(36) and (37) and they are also scaled by (38).

Again, for a quadratic function, Shanno proved in [17] that his algorithms reduce to the Beale's restarting CG-algorithm provided, ELS are used.

3.2. Variable-Storage Conjugate-Gradient Methods

Two new interesting CG-methods, developed in the 1983, are the VM-CG method of Buckley and LeNir [6] and the memoryless QN-method of Shanno. These algorithms both require storage of $O(n)$and in theory and in practical tests converge faster than the standard CG-methods.

Buckley and LeNir [6] mixed CG and VM method is based on the idea of variable storage requirement and on Shanno's idea of modifying a standard CG direction into QN-like form. Their algorithm can be viewed as an extension of Shanno's double update algorithm to a multiple update algorithm [18].

Buckley showed that Shanno's method is more efficient than his earlier algorithm. We therefore consider modifying Shanno's double update algorithm which is still the basis for the Buckley-LeNir algorithm.

3.3. A New Double Update Memoryless Variable Metric Algorithm

In view of the dual CG-VM relationship we observe that a new double update memoryless VM-algorithm can be obtained by using the new family of updates defined in (19). We observe that the standard CG-method can be obtained from (19) provided that the approximation to the inverse Hessian is indeed updated at every step but always from the identity matrix so that the matrix $H\prime_t$ becomes

$$H_t' = 1 - \frac{v_t y_t + y_t v_t^T}{v_t^T y_t} + \frac{2 y_t^T y_t (v_v v_t^T)}{(v_t^T y_t)^2}. \tag{40}$$

Hence

$$H_{k+1}' = H_t' - \frac{v_k y_k^T H_t' + H_t' y_k v_k^T}{v_k^T y_k} + \frac{2(y_k^T H_t' y_k)(v_k v_k^T)}{(v_k^T y_k)^2}. \tag{41}$$

So that the new search direction, in this case is defined by:

$$d_{k+1} = -H'_{k+1}g_{k+1} \tag{42}$$

$$= H'_t g_{k+1} + \frac{v_k^T g_{k+1}}{v_k^T y_k} H'_t y_k - (\frac{2y_k^T H'_t y_k}{v_k^T y_k} - \frac{y_k^T H'_t g_{k+1}}{v_k^T y_k}) v_k, \tag{43}$$

where the two additional vectors are defined by

$$H'_t g_{k+1} = g_{k+1} - \frac{v_t^T g_{k+1}}{v_t^T y_t} y_t + (\frac{2(y_t^T y_t)(v_t^T g_{k+1})}{(v_t^T y_t)^2} - \frac{v_t^T g_{k+1}}{v_t^T y_t}) v_t$$

and

$$H'_t y_k = y_k - \frac{v_t^T y_k}{v_t^T y_t} y_t + (\frac{2(y_t^T y_t)(v_t^T y_k)}{(v_t^T y_t)^2} - \frac{y_t^T y_k}{v_t^T y_t}) v_t. \tag{44}$$

Theorem 1. *The new algorithm reduces to Beale's method for quadratic functions and with ELS.*

Proof. To show that we proceed as follows: From (42) and assuming ELS, we obtain

$$d_{k+1} = -H'_t g_{k+1} + [(y_k^T H'_t g_{k+1})/(v_k^T y_k)] v_k.$$

We also note that for a quadratic function

$$g_{k+1} = Ax_{k+1} + b = A(x_{t+1} + \sum_{i=t+1}^{k} v_i) + b = g_{t+1} + \sum_{i=t+1}^{k} Av_i. \tag{45}$$

Hence, for ELS on a quadratic function

$$v_t^T g_{k+1} = v_t^T g_{k+1} = 0, \tag{46}$$

for $i = t+1, \ldots, k$.

Thus, from (45), we have

$$H'_t g_{k+1} = g_{k+1} - [y_t^T g_{k+1} / v_t^T y_t] v_t. \tag{47}$$

Substituting (47) in (45) we get

$$d_{k+1} = -g_{k+1} + \frac{y_t^T g_{g+1}}{v_t^T y_t} v_t + \frac{y_k^T g_{k+1}}{v_k^T y_k} v_k - \frac{(y_t^T g_{k+1})(y_k^T v_t)}{(v_k^T y_k)(v_t^T y_t)} v_k. \tag{48}$$

Since the conjugacy ensures that:

$$y_k^T v_t = 0, \tag{49}$$

then (48) reduces to

$$d_{k+1} = -g_{k+1} + \frac{y_t^T g_{k+1}}{v_t^T y_t} v_t + \frac{y_k^T g_{k+1}}{v_k^T y_k} v_k. \tag{50}$$

Thus, the search direction (50) is identical to the search direction of Beal's method, defined in (23), and hence the proof is complete.

This new algorithm is implemented such that the direction d_{k+1} is in fact defined by (21) at each restart step, and no further scaling is used.

But for each non-restart step d_{k+1} is scaled according to (38). Hence, the same restarting criterion is employed as in Shanno's algorithms so that their respective practical performances are readily compared.

3.4. Line Search Criterion

For Shanno's memoryless QN-algorithm it is necessary to ensure that

$$v_k^T y_k > 0 \tag{51}$$

for $k > t$ in order to maintain positive definite updating, and hence that a downhill direction d_{k+1} is obtained. However, condition (51) is, in practice, generally replaced by a slightly stronger line search criterion, namely that line searches are terminated when both

$$\mid d_k^T g_{k+1} \mid < \rho_1 \mid d_k^T g_k \mid$$

and

$$f_{k+1} - f_k < \rho_2 v_k^T g_k, \tag{52}$$

where these conditions are sufficient to ensure convergence of any descent method, as quoted in [18]. For all his algorithms, Shanno found that $\rho_2 = 0.0001$ works satisfactorily, but ρ_1 is the critical and sensitive parameter. He found that $\rho_1 = 0.1$ was best for algorithms (13) and (17) and we therefore use the same two values for our new algorithm defined in (45).

4. Numerical Results

In order to assess the performance of the new algorithm (MNEWH), five CG-algorithms are tested on a collection of thirty varied dimensionality test problems. All the algorithms use exactly the same linear search strategy, which is a cubic fitting technique. All algorithms (except the second version of Hestenes-Stiefel CG-method) in this paper use the initial search direction (39) as defined by Shanno in [17]. The comparative performance of the algorithms are evaluated by considering both the total number of function evaluations (NOF) and the total number of iterations (NOI). Whereas NOF is the best measure of actual work done it is dependent on the linear search and the accuracy required; NOI is preferred by some authors for this reason, but the requirement for higher accuracy (and so high NOF) can even reduce NOI. Both should therefore be taken into account.

Five methods are tested: (i) the Shanno CG-method defined by (13) (MBFGS), (ii) the Shanno CG-method defined by (17) (MOREN), (iii) the new CG-method defined by (42) (MNEWH), and (iv) the first version of Hestenes-Stiefel CG-method (HS1). Each of these is restarted every n iterations or whenever (26) is satisfied, with

$$d_{k+1} = -g_{k+1}[(d_k^T d_k)/(g_{k+1}^T g_{k+1})]. \tag{53}$$

The fifth algorithm is the second version of Hestenes-Stiefel CG-method (HS2) which restarts with $-g$ after every n iterations. As may be expected the HS2 method becomes increasingly inefficient as dimensionality n is increased, but the complete results for these five algorithms are presented in the table below.

Analysis of this table shows that the MNEWH method has a clear advantage over Shanno's methods, especially when compared with the MOREN method where it saves overall about 58% in NOF; however, it only saves overall about 8% in NOF over the MBFGS method so that we confirm Shanno's conclusions in [17] about the respective performance of his own two methods. In considering NOI, all the three algorithms perform about the same, though MNEWH has slight advantage over the other two. The table also confirms that the MNEWH algorithm is indeed superior to the standard Hestenes-Stiefel CG-method for both versions.

In conclusion, we have considered a memoryless new CG algorithm, which is clearly superior to Perry's algorithm and Shanno's MOREN method (on the same convergence criterion). The $O(n)$ double-updating MOREN method should be usually the first choice when the $O(n^2)$ QN methods just require backing store; however, the number of vectors required in $O(n)$ methods must become significant when back storing again becomes necessary because of the dimensionality of the problem. Based on our experience with these method, the preferred methodsfor increasing n are in succession: QN methods, MNEWH, MOREN and the standard CG. There is therefore no single best method to be recommended for all n.

Function	N	HS1	HS2	MBFGS	MOREN	MNEWH
		NOI(NOF)	NOI(NOF)	NOI(NOF)	NOI(NOF)	NOI(NOF)
ROSEN	2	30(73)	34(87)	30(73)	34(170)	31(75)
CUBIC	2	17(50)	19(53)	17(50)	19(128)	17(50)
BEALE	2	10(27)	10(26)	10(27)	10(43)	10(27)
BOX	2	11(57)	6(47)	11(57)	11(61)	11(57)
FREUD	2	10(21)	8(22)	10(21)	10(53)	10(21)
BIGGS	3	16(60)	14 (42)	19 (65)	12(42)	17(60)
RECIPE	3	5(19)	5(16)	5(19)	5(21)	5(19)
HELICAL	3	37(81)	23(53)	36(77)	29(127)	46(99)
POWL3	3	17(39)	18(40)	17(41)	14(48)	16(37)
POWELL	4	63(174)	65(170)	40(105)	59(277)	29(75)
WOOD	4	27(62)	26(60)	23(54)	23(83)	23(51)
DIXON	10	23(49)	20(43)	23(49)	23(69)	23(49)
OREN	10	14(61)	10(45)	13(55)	14(52)	14(60)
EXPOWELL	20	58(175)	60(162)	62(177)	42(174)	40(109)
EXWOOD	20	22(54)	47(102)	25(59)	25(103)	26(57)
NONDIGN	20	27(67)	19(52)	27(67)	27(134)	24(56)
SUMQUAR	25	8(38)	8(44)	9(40)	8(31)	8(38)
OREN	30	29(102)	18(71)	25(95)	25(76)	27(96)
TRIDIGN	30	30(62)	30(61)	30(62)	31 (91)	31(64)
SHALLOW	40	8(25)	9(21)	8(25)	8(31)	8(25)
FULL	40	44(91)	44(89)	46(95)	46(134)	46(95)
OREN	50	33(108)	24(90)	36(138)	35(115)	34 (123)
EXROSEN	60	27(74)	17(52)	27 (74)	29(136)	32(91)
EXWOOD	60	34(82)	65(138)	32(76)	34(125)	33(82)
WOLFE	80	49(99)	49(99)	49(99)	49(147)	49(99)
NONDIGN	90	27(69)	22(60)	27(69)	27(155)	24 (58)
EXROSEN	100	30(75)	17 (52)	27(68)	29 (128)	30(74)
EXPOWELL	100	46(123)	105(276)	66(207)	46(201)	41(111)
EXWOOD	100	35(82)	103(213)	32(76)	36(140)	39(91)
EXROSEN	1000	27(70)	18(55)	27(70)	30(127)	33(82)

Total	**HS1**	**HS2**	**MBFGS**	**MOREN**	**MNEWH**
NOI	814	913	809	790	777
NOF	2167	2341	2190	3222	2031

Overall comparative performance

	MNEWH	**HS1**	**HS2**	**MBFGS**	**MOREN**
NOI	100	104.7	117.5	104.1	101.6
NOF	100	106.6	115.2	107.8	158.6

All the algorithms terminate when $||g_{k+1}||_2 < 10^{-7}$.

References

[1] E.M.L. Beale, *A Derivation of Conjugate Gradients, Numerical Methods for Nonlinear Optimization,* Ed. F.A. Lootsma, Academic Press, London (1972).

[2] M.C. Biggs, Minimization algorithms making use of non-quadratic properties of the objective function, *J. of the Institute of Mathematics and its Applications*, **8** (1971), 315-327.

[3] M.C. Biggs, A note on minimization algorithms which make use of non-quadratic properties of the objective function, *J. of Institute of Mathematics and its Applications*, **12** (1973), 337-338.

[4] K.W. Brodlie, *Some Topics in Unconstrained Minimization*, Ph.D. thesis, University of Dundee (1973).

[5] 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.

[6] A. Buckley and A. LeNir, Quasi Newton-like variable storage conjugate gradients, *Mathematical Programming*, **27** (1983), 155-175.

[7] L.C.W Dixon, Conjugate gradient algorithm quadratic termination without line searches, *J. of the Institute of Mathematics and its Applications*, **15** (1975).

[8] R. Fletcher and C. M. Reeves, Function minimization by conjugate gradients, *Computer Journal*, **7** (1964), 149-154.

[9] H.R. Hestenes and E. Stiefel, Methods of conjugate gradients for solving linear systems, *J. of Research of the National Bureau of Standards*, **49** (1952), 409-436.

[10] G.P. McCormick and K. Ritter, Methods of conjugate directions versus quasi-Newton methods, *Mathematical Programming*, **3** (1972), 101-116.

[11] L. Nazareth, A relationship between BFGS and conjugate-gradient algorithms and its implementations for new algorithms, *SIAM J. on Numerical Analysis*, **16** (1979), 794-800.

[12] S.S. Oren, Self-scaling variable metric algorithm, Part II, *Management Science*, **20** (1974), 863-874.

[13] S.S. Oren, On the selection of parameters in self-scaling variable metric algorithms, *Mathematical Programming*, **3** (1974), 351-367.

[14] S.S. Oren and E. Spedicato, Optimal conditioning of self-scaling variable metric algorithms, *Mathematical Programming*, **10** (1976), 70-90.

[15] A. Perry, A modified conjugate gradient algorithm, *Operations Research*, **26** (1978), 26-33.

[16] M. J. D. Powell, Some convergence properties of the conjugate gradient method, *Mathematical Programming*, **11**, No. 1 (1976), 42-49.

[17] D.F Shanno, Conjugate gradient methods with inexact searches. *Math. Oper. Res.*, **3** (1978), 244-256.

[18] D. Shanno and G. Phua, Inexact step lengths and quasi-Newton methods, *Information Processing*, **77** (Ed. B. Gilchrist), North Holland (1977), 189-193.