A New Variable -Metric Conjugate Gradient Algorithm

ISSAM A.R. MOGHRABI
MIS Department, College of Business
Gulf University for Science and Technology
KUWAIT
Moughrabi.i@gust.edu.kw

Abstract— Motivated by the success of Shanno's memoryless Conjugate Gradient (CG) methods [28,29], this paper derives three new scaled quasi-Newton like CG algorithm that utilize an update formula that is invariant to a scaling of the objective function. The computation of the search directions, at each iteration, is done in two steps. The aim of developing such self scaling Variable Metric CG methods is to improve the quality of the generated search direction vectors. The computations involved are rather cheap as they merely involve a number of inner products and require just extra O(n) storage requirements. The extra requirements are shown to pay off when the algorithm is numerically compared to that developed by Shanno.

Keywords— Unconstrained Optimization, Conjugate Gradient Methods, Variable Metric methods, Inexact Line Search

1 Introduction

Conjugate Gradient (CG) methods were initially developed by Fletcher and Reeves [12] and were used to solve general unconstrained minimization problems. Those methods are still favored to the more rapidly convergent Quasi-Newton methods (QN) for big problems due to their low storage requirements (O(n)), instead of $O(n^2)$, as is the case with QN methods. This constitutes enough justification for constantly attempting to improve these methods. In this work the focus is rather on methods which combine the merits of both the CG and QN methods, as was done earlier by Perry [23] and Shanno [28,29]. Our derivation exploits the success of the multi-step QN methods [18,19] to derive a CG algorithm that utilizes data available from recent iterations so that convergence is numerically accelerated further. Section 2 of this paper briefly introduces the CG methods and variants. Section 3 presents the idea of memoryless self scaling Variable-Metric CG methods. Section 4 focuses on the derivation of the new algorithms. Then, the numerical results are discussed in Section 4.

2 Memoryless Self-Scaling Variable-Metric Conjugate Gradient Methods

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

$$d_i^T A d_j = 0, \ \forall i \neq j. \tag{1}$$

CG- methods are iterative and generate a sequence of approximations to the minimum x^* of a scalar function $f(x)$ in order to solve

$$minmize\ f(x), x \in R^n, where\ f: R^n \to R.$$

The sequence x_i is defined by the following recurrence

$$d_i = \begin{cases} -g_i, & for\ i=0, \\ -g_i + \beta_{i-1} d_{i-1}, & for\ i \geq 1, \end{cases} \tag{2}$$

where g_i is the gradient of $f(x_i)$, β_i is a positive scalar chosen to minimize $f(x)$ along the search direction d_i, and the standard definition of β_i is given by

$$\beta_i = \frac{y_i^T g_{i+1}}{d_i^T y_i}, \tag{3}$$

for

$$y_i = g_{i+1} - g_i. \tag{4}$$

The definition of β_i in (3) is the one given in Hestenes and Stiefel [17] and is a modification to the original CG method derived by Polak and Ribiére [24] and Polyak [25].

This choice for the scalar β_i is such that to make the search vectors d_k and d_{k+1} conjugate when the line searches are exact (ELS). However, since in practice line searches are not exact, Perry [23] rewrote (2) under the assumption of inexact line searches (ILS) as follows

$$d_{i+1} = -\left[I - \frac{s_i y_i^T}{y_i^T s_i} + \frac{s_i s_i^T}{y_i^T s_i} \right] g_{i+1}, \tag{5}$$

$$= Q_{i+1} g_{i+1} \tag{6}$$

for

$$s_i = x_{i+1} - x_i. \tag{7}$$

The matrix Q_{i+1} satisfies a relation similar to the Secant relation, namely

$$Q_{i+1}{}^T y_i = s_i. \tag{8}$$

Perry's method [23] performance is only slightly better than the standard CG-method. Besides, the matrix Q_{i+1} is not necessarily symmetric or positive definite so that the direction vector in (7) may not be a descent direction.
Shanno [28] derived a similar algorithm that satisfies the Secant relation

$$H_{i+1}{}^{T} y_i = s_i, \tag{9}$$

where H_{i+1} is an approximation to the inverse Hessian. Shanno proposed

$$Q_{i+1} = I - \frac{s_i y_i^T + y_i s_i^T}{y_i^T s_i} + [1 + \frac{y_i^T y_i}{y_i^T s_i}] \frac{s_i s_i^T}{y_i^T s_i}. \tag{10}$$

Update formula (10) is actually the BFGS update formula with the update applied to the identity matrix at each iteration.

In fact, a similar dual relationship to (9)-(10) can be exhibited for any member of Broyden's θ-class update [5,8,20].

The CG-method for which the search direction is computed using

$$d_{i+1} = -Q_{i+1}{}^{T} g_{i+1} \tag{11}$$

is referred to as a memoryless BFGS method.

The idea of self-scaling was originally developed by Oren [21] and Oren & Spedicato [22]. Oren modified in [21] the Broyden ϑ-class of updates as follows

$$H_{i+1} = \left[H_i - \frac{H_i y_i y_i^T H_i}{y_i^T H_i y_i} + \vartheta_i r_i r_i^T \right] \mu_i + \frac{s_i s_i^T}{y_i^T s_i},$$

where

$$r_i = \sqrt{y_i^T H_i y_i} \left(\frac{s_i}{y_i^T s_i} - \frac{H_i y_i}{y_i^T H_i y_i} \right)$$

and the specific value chosen for ϑ_i results in different update formula that belong to the Broyden family. For example, the BFGS update corresponds to $\vartheta_i = 1$. The scaler μ_i is defined by (see [21])

$$\mu_i = \frac{y_i^T s_i}{y_i^T H_i y_i}.$$

Shanno [28] used this formula for H_{i+1} to derive a modified CG-method with H_i replaced by the identity matrix to eliminate the need for storing any matrices. This resulted in the following memoryless search direction formula

$$d_{i+1} = -\left[\frac{y_i^T s_i}{y_i^T y_i} g_{i+1} + \left(\frac{2 g_{i+1}^T s_i}{s_i^T s_i} - \frac{g_{i+1}^T y_i}{y_i^T y_i} \right) s_i - \frac{g_{i+1}^T s_i}{y_i^T y_i} y_i \right].$$

This choice is equivalent to scaling the memoryless BFGS by by μ_i.

The results given by this modified CG-method in [27] were rather disappointing.

Another memoryless Variable Metric (VM) method can be obtained by scaling, at each iteration, the update

$$H_{i+1} = \left[H_i - \frac{H_i y_i y_i^T H_i}{y_i^T H_i y_i} + \vartheta_i r_i r_i^T \right] \mu_i + \frac{s_i s_i^T}{y_i^T s_i},$$

by a positive scalar σ_k given by [27]

$$\sigma_i = \frac{1}{\mu_i}.$$

The parameter σ_i has the advantage of making the sequence of iterates invariant under multiplication of the objective function by a constant scalar. This results in the following memoryless VM search direction

$$d_{i+1} = -g_{i+1} - \left[\left(\frac{2 y_i^T y_i g_{i+1}^T s_i}{(y_i^T s_i)^2} - \frac{g_{i+1}^T y_i}{y_i^T s_i} \right) s_i + \frac{g_{i+1}^T s_i}{y_i^T s_i} \right] y_i \tag{12}$$

For exact line search, we have $g_{i+1}^T s_i = 0$ and hence d_{i+1} becomes

$$d_{i+1} = -g_{i+1} + \frac{g_{i+1}^T y_i}{y_i^T s_i} d_i,$$

which is the standard Hestenes & Stiefel CG-method [17] and therefore has n-step convergence to the minimum of a quadratic function. Thus the CG-method is defined precisely by this new VM update (12), where the approximation to the inverse Hessian is reset to the identity matrix at every step.

The CG-algorithm defined by (2)-(4) exhibit a linear rate of convergence unless the method is restarted (generally every n steps) with direction $d_t = -g_t$, (see Powell [26], Biggs [3,4]). Powell suggests a restart every n steps or whenever

$$|g_{i+1}{}^{T} g_i| \geq 0.2 |g_{i+1}{}^{T} g_{i+1}|.$$

Since the step taken in the direction of the negative gradient frequently results in a considerably small reduction in the objective function, Beale [2] derived a restart criterion intended to improve convergence rate. The restart step was taken to be the computed direction d_t rather than $-g_t$. Subsequent non-restart steps are defined by (t being the index of the latest restart step):

$$d_{i+1} = -g_{i+1} + \beta_i d_i + \gamma_i d_t,$$

where

$$\beta_i = (y_i{}^{T} g_{i+1}) / (d_i{}^{T} y_i)$$

and

$\gamma_i = (y_t^{\mathrm{T}} g_{i+1}) / (d_t^{\mathrm{T}} y_t)$,

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

Shanno [28], inspired by Beale's approach [2], proposed

$$d_{i+1}=-[I-\frac{d_i y_i^T}{y_i^T s_i}+\frac{d_t y_t^T}{y_t^T s_t}]\, g_{i+1} \tag{13}$$

where y_i is given in (4), s_i is as in (7) and t is the index of the last restart. The update matrix uses information from two prior points, namely x_i and x_t where the information gathered at x_t is critical and must be exploited. Shanno [28] defined, for $k > t$, the following double update scheme

$$H_t=I-\frac{s_t y_t^T+y_t s_t^T}{y_t^T s_t}+[1+\frac{y_t^T y_t}{y_t^T s_t}]\frac{s_t s_t^T}{y_t^T s_t} \tag{14}$$

and

$$H_{i+1}=H_t-\frac{s_i y_i^T H_t+H_t y_i s_i^T}{y_i^T s_i}+[1+\frac{y_i^T H_t y_i}{y_i^T s_i}]\frac{s_i s_i^T}{y_i^T s_i} \tag{15}$$

The search direction at iteration i is computed using

$$d_{i+1}=-H_{i+1}g_{i+1}.$$

$$d_{i+1} = -H_{i+1}g_{i+1}$$
$$= -H_t g_{i+1} + \frac{s_i^T g_{i+1}}{s_i^T y_i} H_t y_i - ((1+\frac{y_i^T H_t y_i}{s_i^T y_i})\frac{s_i^T g_{i+1}}{s_k^T y_k} - \frac{y_i^T H_t g_{i+1}}{s_i^T y_i}) s_i$$

The vector $H_t g_{i+1}$ and $H_t y_i$ are defined by

$$H'_t\, g_{i+1} = g_{i+1} - \frac{v_t^T g_{i+1}}{v_t^T y_t} y_i + ((1+\frac{y_t^T y_t}{v_t^T y_t})\frac{v_t^T g_{i+1}}{v_t^T y_t} - \frac{y_t^T g_{i+1}}{v_t^T y_t}) v_t$$

and

$$H'_t\, y_t = y_t - \frac{v_t^T y_i}{v_t^T y_t} y_t + ((1+\frac{y_t^T y_t}{v_t^T y_t})\frac{v_t^T y_i}{v_t^T y_t} - \frac{y_t^T y_i}{v_t^T y_t}) v_t.$$

In implementing this algorithm additional storage is required to store vectors x_{i+1}, x_i, g_{i+1}, g_i, d_i, d_t, and y_t (a total storage still of order n).

Another far more successful search direction, proposed by Shanno, is generated by using update (15) in the computation of the search direction. This yield for $i>t$

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

Shanno therefore suggested scaling the matrix H'_t with

$$\eta_t = (v_i^T y_i) / (y_t^T y_t)$$

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

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

and

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

However, he also tested the application of the Flecher [15] scaling in his numerical trials. In consequence he proposed using only the scaled H'_t at restart steps, and at each non-restart step to scale according to the following Flecher scaling criterion:

$$d'_{i+1} = [2(f_{i+1} - f_i)/d_{i+1}^T g_{i+1}] d_{i+1} \ .$$

The vectors that require retention in storage for this method are x_{i+1}, x_i, g_{i+1}, g_i, d_i, d_t, and y_t.

3 A new Memoryless Variable Metric CG Algorithm

In this derivation, we start by presenting a new variable metric (VM) update formula that will be used in the derivation of the new memoryless VM CG method. The notion of self-scaling quasi-Newton algorithms was first proposed by Oren [21], and Oren & Spedicato [22]. The update formula used in this paper is a generalization of Oren's update [21], modified to satisfy

$$H_{i+1}u_i=\sigma_i v_i, \tag{16}$$

for $\sigma_i=\frac{u_i^T H_i u_i}{u_i^T v_i}$, for some vectors u_i and v_i (35)

The resulting update formula is given by

$$H_{i+1}=H_i+\frac{2u_i^T H_i u_i}{(u_i^T v_i)^2} v_i v_i^T - \frac{v_i u_i^T H_i + H_i u_i v_i^T}{u_i^T v_i}. \tag{17}$$

The parameter σ_i is introduced to make the method invariant under multiplication of the objective function by a constant. It is worth mentioning that the standard CG-method can be obtained from (17) if the approximation to the inverse Hessian H_i is taken to be the identity matrix and the vectors u_i and v_i are chosen to be y_i and s_i, respectively. Like in (14), the matrix H_t is this given as

$$H_t=I+\frac{2u_t^T u_t}{(u_t^T v_t)^2} v_t v_t^T-\frac{v_t u_t^T+u_t v_t^T}{u_t^T v_t}.$$

$$H_t=I+\frac{2u_t^T u_t}{(u_t^T v_t)^2} v_t v_t^T-\frac{v_t u_t^T+u_t v_t^T}{u_t^T v_t}. \quad (18)$$

Consequently,

$$H_{i+1}=H_t+\frac{2u_i^T H_t u_i}{(u_i^T s_i)^2} s_i s_i^T-\frac{s_i u_i^T H_t+H_t u_i s_i^T}{u_i^T s_i}. \quad (19)$$

The search direction, in this case, is defined by:

$$\begin{aligned} d_{i+1}&=-H_{i+1}g_{i+1}\\ &=H_t g_{i+1}+\frac{v_i^T g_{i+1}}{u_i^T v_i} H_t u_i-\frac{2u_i^T H_t u_i-u_i^T H_t g_{i+1}}{u_i^T v_i} v_i, \end{aligned} \quad (20)$$

where the two additional vectors that need to be stored to implement this method are defined by

$$H_t g_{i+1}=g_{i+1}-\frac{s_t^T g_{i+1}}{y_t^T s_t} y_t+\frac{(2y_t^T y_t)(s_t^T g_{i+1})-y_t^T g_{i+1}}{y_t^T s_t} s_i \quad (21)$$

and

$$H_t y_i=y_i-\frac{y_i^T v_t}{y_t^T v_t} y_t+\left(\frac{2y_t^T y_t+v_t^T y_i}{(y_t^T v_t)^2}-\frac{y_t^T y_i}{y_t^T v_t}\right) v_t. \quad (22)$$

The following theorem reveals that the search direction (20) is a special case of the Beale's 3-term CG method, given as

$$d_{i+1}=-g_{i+1}+\beta_i d_i+\gamma_i d_t, \quad (23)$$

where

$$\beta_i=(y_i^T g_{i+1})/(d_i^T y_i)$$

and

$$\gamma_i=(y_t^T g_{i+1})/(d_t^T y_t) \quad ,$$

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

Theorem 1: If $f(x)$ is quadratic and the vectors u_i and v_i are chosen to be y_i and s_i, respectively, the search direction given by (20) is equivalent to the 3-term search directions generated by Beale's method [2] when the line searches are exact.

Proof. From (20), and since an exact line search is used gives $s_i^T g_{i+1}=0$, we obtain

$$d_i=-H_t g_{i+1}+\frac{y_i^T H_t g_{i+1}}{s_i^T y_i} s_i.$$

We also note that for a quadratic function

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

Hence, for ELS and quadratic functions

$$s_t^T A s_k=0, \text{ for } i=t+1,\ldots,i.$$

Thus,

$$s_t^T g_{i+1}=s_t^T g_{t+1}=0. \quad (24)$$

Now from (21), using (24), we have

$$H'_t g_{i+1}=g_{i+1}-[y_t^T g_{i+1}/v_t^T y_t]v_t. \quad (25)$$

Substituting (25) in (23) we get

$$d_{i+1}=-g_{i+1}+\frac{y_t^T g_{i+1}}{s_t^T y_t} s_t+\frac{y_i^T g_{i+1}}{v_i^T y_k} s_i-\frac{(y_t^T g_{i+1})(y_i^T s_t)}{(s_i^T y_i)(s_t^T y_t)} s_i. \quad (26)$$

By the conjugacy condition we have

$$y_i^T s_t=0.$$

Thus, (26) reduces to

$$d_{i+1}=-g_{i+1}+\frac{y_t^T g_{i+1}}{s_t^T y_t} s_t+\frac{y_i^T g_{i+1}}{s_i^T y_i} s_i. \quad (27)$$

The search direction (27) is identical to the search direction of Beale's method (23) (see [2]) and hence the proof is complete.

Usually, conjugate gradient algorithms are periodically restarted. Powell's restarting procedure [26], used in this algorithm, is for testing whether there is very little orthogonality left between the current gradient and the previous one. At step r when

$$|g_{r+1}^T g_r|\geq 0.2\|g_{r+1}\|^2.$$

the algorithm may be restarted using (18) with H_t set to I. For $i = r+1$, (20) is employed in the computation of the search direction.

4 Choosing the vectors u_i and v_i

One obvious choice for the vectors is $u_i = y_i$ and $v_i = s_i$. This choice defines our first algorithm and for which a variant of the standard Secant equation is satisfied, namely

$$H_{i+1}y_i=\sigma_i s_i, \quad \textbf{(MNEWH1)}$$

for

$$\sigma_i = \frac{y_i^T H_i y_i}{s_i^T y_i}.$$

The other choice is inspired by the approach of Wei et al. [30]. Assuming that the objective function is smooth, then using the Taylor series for $f(x)$ at the point x_{i+1} gives

$$f(x) \cong f_{i+1} + g_{i+1}^T(x - x_{i+1}) + \frac{1}{2}(x - x_{i+1})^T G_{i+1}(x - x_{i+1}),$$

where G_{i+1} denotes the Hessian of $f(x)$ at the point x_{i+1}.
It follows that

$$f_i \cong f_{i+1} - s_i^T g_{i+1} + \frac{1}{2} s_i^T G_{i+1} s_i \ .$$

Thus,

$$s_i^T B_{i+1} s_i \ = 2(f_i - f_{i+1}) + s_i^T(g_i + g_{i+1}) + s_i^T y_i.$$

However, if the following special case Taylor series for $f(x)$ is used

$$f_i(x) \cong f(x) + \frac{1}{2}(x - x_i)^T A_i (x - x_i),$$

for A_i being a symmetric positive definite matrix, and using the relationship obtained for $s_i^T B_{i+1} s_i$, then the following modified version of the Secant equation is obtained

$$B_{i+1} s_i = \hat{y}_i, \tag{28}$$

for

$$\hat{y}_i = y_i + \frac{\rho_i}{\|s_i\|^2} s_i,$$

where $\rho_i = 2(f_i - f_{i+1}) + (g_i + g_{i+1})^T s_i$.

Adopting this choice in our derivation, we have from (16) and (28)

$$H_{i+1}\hat{y}_i = \sigma_i s_i, \qquad \textbf{(MNEWH2)}$$

for the choices $u_i = \hat{y}_i$ and $v_i = s_i$.

The last choice is based on the modified secant equation derived in [14,18,19]. The methods introduced by the authors utilize the step vectors s_i and s_{i-1} (along with the corresponding y_i and y_{i-1}) in the construction of a variant to the standard quasi-Newton methods that are based on the classical secant equation. The idea is that previous iteration data is discarded after used once and that exploiting that data in the construction of the Hessian (or its inverse) approximation at each iteration pays off, as indicated by the results presented for the multi-step methods. The inverse Hessian approximation update generally satisfies:

$$H_{i+1}(y_i - \mu_{i-1} y_{i-1}) = s_i - \mu_{i-1} s_{i-1} \qquad \textbf{(MNEWH3)}$$

or

$$v_i = H_{i+1} u_i$$

where

$$\mu_{i-1} = \frac{\delta_{i-1}^2}{2\delta_{i-1} + 1}$$

and

$$\delta = \frac{\|s_i\|}{\|s_{i-1}\|}.$$

This expression for δ may be generalized by introducing a scaling factor, $\gamma \geq 0$ (see [18]) that provides a control mechanism for convenient easy switching to the standard secant equation update method obtained by setting the scalar γ to zero. Therefore,

$$\delta = \gamma \frac{\|s_i\|}{\|s_{i-1}\|}.$$

5 Numerical Results and Conclusion

For the above memoryless QN-algorithms it is necessary to ensure that

$$u_i^T v_i > 0 \quad \text{for } i \geq 0, \tag{29}$$

so that the positive definiteness of the update matrix is maintained to ensure that the computed search vector is downhill. However, condition (29) is, in practice, replaced by a stronger line search criteria as follows

$$| s_i^T g_{i+1} | < \rho_1 | s_i^T g_i | \tag{30}$$

and

$$f_{i+1} - f_i < \rho_2 s_i^T g_i. \tag{31}$$

Conditions (30) and (31) are sufficient to ensure convergence of any descent method [16]. Shanno used ρ_1=0.1 and ρ_2=0.0001 and to test his double update algorithms and we use those same choices for our methods.
In order to assess the performance of the new algorithms (MNEWH1, MNEWH2, MNEWH3), those are benchmarked against Shanno's memoryless QN algorithm [28]. The methods are tested on a collection of 30 varied dimensionality test problems with dimensions ranging $2 \leq n \leq 1000$. The test functions can be found in [13,14,15,18]. A cubic fitting technique line search strategy is used for all methods. The algorithms terminate when the gradient vector magnitude is less than 10^{-5}. All methods are restarted every n iterations or whenever (12) is satisfied, with

$$d_{i+1} = \frac{d_i^T d_i}{g_{i+1}^T g_{i+1}} d_i.$$

Performance of the algorithms is evaluated by considering both the total number of function evaluations (NOF) and the total number of iterations (NOI). The results are reported in Table 1.

Analysis of this table shows that the three derived methods have a clear advantage over Shanno's method. When it comes to function/gradient evaluations, MNEWH3 seems to perform best (as it saves an overall about 28.4% in NOF). However, it only saves overall about 4.81% in NOI). When it comes to the number of iterations (NOI), MNEWH2 is the winner (as it saves an overall about 16.8% in NOF and saves overall about 8.6% in NOI).

In conclusion, the new double update methods developed here are promising especially when the number of function/gradient evaluations is of significance. The number of extra vectors required is O(n). Specifically, the number of vectors required to keep in storage seems to be a reasonable tradeoff against the numerical gains incurred.

TABLE I. NUMERICAL COMPARISON OF THE ALGORITHMS

function	Size (n)	MSHANO [a]	MNEWH 1 [a]	MNEWH 2	MNEWH 3
		NOI(NOF)	*NOI(NOF)*	*NOI(NOF)*	*NOI(NOF)*
ROSEN	2	34(170)	31(105)	32(144)	36(141)
CUBIC	2	19(128)	17(90)	21(111)	20(112)
BEALE	2	10(43)	10(27)	11(39)	12(41)
BOX	2	11(61)	11(57)	11(58)	11(57)
FREUD	2	10(53)	10(29)	9(31)	11(29)
BIGGS	3	12(42)	17(60)	16(51)	17(54)
RECIPE	3	5(21)	5(19)	4(20)	5(19)
HELICAL	3	29(127)	46(99)	33(101)	31(103)
POWL3	3	14(48)	16(37)	15(35)	15(36)
POWELL	4	59(277)	29(175)	31(181)	33(179)
WOOD	4	23(83)	23(51)	23(55)	24(51)
DIXON	10	23(69)	23(49)	23(48)	23(48)
OREN	10	14(52)	14(60)	14(58)	16(55)
EX-POWELL	20	42(174)	40(109)	41(104)	39(106)
EX-WOOD	20	25(103)	26(57)	25(66)	26(57)
NON-DIGN	20	27(134)	24(56)	25(78)	27(51)
SUM-QUAR	25	8(31)	8(38)	8(34)	9(30)
OREN	30	25(76)	27(96)	26(80)	26(71)
TRI-DIGN	30	31 (91)	31(64)	30(61)	31(60)
SHALLOW	40	8(31)	8(25)	7(23)	6(22)
FULL	40	46(134)	46(95)	44(94)	44(91)
OREN	50	35(115)	34 (133)	34(128)	33(111)
EX-ROSEN	60	29(136)	32(91)	30(95)	28(88)
EX-WOOD	60	34(125)	33(82)	33(82)	31(78)
WOLFE	80	49(147)	49(99)	48(101)	44(90)
NON-DIGN	90	27(155)	24 (58)	27(149)	22(51)
EX-ROSEN	100	29 (128)	30(174)	30(147)	29(111)
EX-POWELL	100	46(201)	41(131)	44(176)	41(119)
EX-WOOD	100	36(140)	39(191)	37(180)	33(130)
EXROSEN	1000	30(127)	33(152)	34(151)	29(116)
TOTAL NOI(NOF)		790(3222)	777(2491)	722(2681)	752(2307)

[a.] algorithms terminate when the gradient vector magnitude is less than 10^{-5}

References

[1] N. Anderi, "A scaled BFGS preconditioned conjugate gradient algorithm for unconstrained optimization", Applied Mathematics Letters, vol. 20, pp. 645-650, 2007.

[2] E.M.L Beale, "A Derivation of Conjugate Gradients," in Numerical Methods for Nonlinear Optimization, 7ed. F.A. Lootsma, London: Academic Press, 1972, pp.94-112.

[3] M.C. Biggs, "Minimization algorithms making use of non-quadratic properties of the objective function", Journal of the Institute of Mathematics and its Applications, vol. 8, pp. 315-327, 1971.

[4] 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 Applications, vol.12, pp. 337-338, 1973.

[5] C.G. Broyden, "The convergence of a class of double-rank minimization algorithms - Part 2: The new algorithm", J. Inst. Math. Applic., vol. 6, pp. 222-231, 1970.

[6] R.H. Byrd, R.B. Schnabel, and G.A. Shultz, "Parallel quasi-Newton methods for unconstrained optimization", Math. Programing, vol. 42 , pp. 273-306, 1988.

[7] Y.H. Dai, Y. Yuan, A nonlinear conjugate gradient method with a strong global convergence property, SIAM J. Optim., vol. 10, pp. 177–182, 1999.

[8] J.E. Dennis and R.B. Schnabel, "Least change secant updates for quasi-Newton methods", SIAM Review, vol. 21, pp. 443-459, 1979.

[9] R. Fletcher, Practical Methods of Optimization (second edition), Wiley, New York, 1987.

[10] R. Fletcher, "A new approach to variable metric algorithms", Comput. J. vol. 13, pp. 317-322, 1970.

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

[12] R.Fletcher, C.Reeves, "Function minimization by conjugate gradients", Computer J., vol. 7, pp. 149–154, 1964.

[13] J.A. Ford and I.A. Moghrabi, "Using function-values in multi-step quasi-Newton methods", J. Comput. Appl. Math., vol. 66 (), pp. 201-211, 1996.

[14] J.A. Ford and I.A. Moghrabi, "Multi-step quasi-Newton methods for optimization", J. Comput. Appl. Math, vol. 50, pp. 305-323, 1994.

[15] J.A. Ford and I.A. Moghrabi, "Alternative parameter choices for multi-step quasi-Newton methods", Optimization Methods and Software , vol. 2, pp. 357-370, 1993.

[16] W.W. Hager, H.C. Zhang, "A new conjugate gradient method with guaranteed descent and an efficient line search", SIAM J.Optim., vol. 16, pp.170–192, 2005.

[17] M.R. Hestenes, E. Stiefel, "Methods of conjugate gradients for solving linear systems", J. Res. Nat. Bur. Stan., Sec. B 48, pp. 409–436, 1952.

[18] I.A.R. Moghrabi, "Numerical experience with multiple update quasi-newton methods for unconstrained optimization", Journal of Mathematical Modeling and Algorithms, vol. 6, pp. 231-238, 2007.

[19] I.A.R. Moghrabi, "Implicit extra-update multi-step quasi-newton methods", Int. J. Operational Research, vol. 28, pp. 69-81, 2017.

[20] L. Nazareth, "A relationship between BFGS and conjugate-gradient algorithms and its implementations for

new algorithms", SIAM Journal on Numerical Analysis, vol.16, pp. 794-800, 1979.

[21] S.S. Oren, "Self-scaling variable metric algorithm without line search for unconstrained minimization", Mathematics of Computation , vol. 27 (), pp. 873-885, 1973.

[22] S.S. Oren and E. Spedicato, "Optimal conditioning of self-scaling variable metric algorithms", Mathematical Programming , vol. l0, pp. 70-90, 1976.

[23] A. Perry, "A modified conjugate gradient algorithm", Oper. Res., vol. 26, pp. 1073-1078, 1978.

[24] E. Polak, G. Ribiére, "Notesurla convergence de directions conjuguées", Rev. Francaise Infomat Recherche Operatonelle, vol. 16 , pp. 35–43, 1969.

[25] B. T. Polyak, "The conjugate gradient method in extreme problems", USSR Comp. Math. Phys, vol.. 9, pp. 94–112, 1969.

[26] M.J.D. Powell, "Restart procedures for the conjugate gradient method", Math. Program., vol. 12, pp. 241–254, 1977.

[27] D.F. Shanno, "Conditioning of quasi-Newton methods for function minimization", Math. Comp., vol. 24, pp. 647-656, 1970.

[28] D.F. Shanno, "Conjugate gradient methods with inexact searches", Math. Oper. Res., vol. 3, pp. 244–256, 1978.

[29] D.F. Shanno, "On the convergence of a new conjugate gradient algorithm", SIAM J. Numer. Anal., vol. 15, pp. 1247–1257, 1978.

[30] Wei, Z., Li, G., Qi, L., New quasi-Newton methods for unconstrained optimization problems, Appl. Math. Comput. 175: 1156-1188 (2006).