Int. J. Open Problems Compt. Math., Vol. 5, No. 1, March 2012
ISSN 1998-6262; Copyright © ICSRS Publication, 2012
www.i-csrs.org

New Scaled Three-Term Memoryless VM-Methods for Solving Nonlinear Unconstraint Problems

Abbas Y. Al-Bayati and Khalil Kh. Abbo

Department of Mathematics, College of Computers Sciences and Mathematics
University of Mosul, Iraq.
e-mail: profabbasalbayati@yahoo.com

Received 13 June 2011
Revised 9 October 2011
Accepted 23 October 2011

Abstract

Four new scaled three-term memoryless VM methods for solving nonlinear unconstrained problems are presented. The basic idea is to deal with Al-Bayati's (1991) and Biggs's (1983) self-scaling VM-updates in the frame of new scaled CG-methods. Birgin-Martinez (2001) and Abbo (2007) positive parameters are used to scale these spectral CG-methods. The new search directions are reset to the standard Steepest Descent (SD) direction when Powell's (1977) restarting criterion holds. Andrei's (2010) acceleration scheme of the step-size parameter has been employed in the new proposed methods to improve the efficiency of such methods. Under common assumptions; the new methods are proved to be globally convergent. Computational results for a set consisting of 100 unconstrained optimization test problems show that the new methods substantially outperforms the scaled memoryless BFGS method.

Key words: Scaled CG Method, Unconstrained Optimization, Convergence Property, Line Searches, Memoryless Self-Scaling VM-Updates, Scaled Memoryless BFGS Method.

AMS (2000) subject classification: 90C06, 90C26.

1. Introduction.

In this study, we consider the unconstrained optimization problem:

$$\min\{f(x)|x\in R^n\}, \tag{1}$$

where $f:R^n\to R$ is a continuously differentiable function, and its gradient at point x_k is denoted by $g(x_k)$, or g_k for the sake of simplicity. n is the number of variables, which is automatically assumed to be large. The iterative formula of nonlinear CG method is given by:

$$x_k = x_{k-1} + \alpha_k d_k, \tag{2}$$

where α_k is a step-length, and d_k is a search direction which is determined by:

$$d_k = \begin{cases} -g_0, & if \quad k=0, \\ -g_k + \beta_k d_{k-1}, & if \quad k\geq 1, \end{cases} \tag{3}$$

where β_k is a scalar. The search direction d_k is generally required to satisfy:

$$g_k^{\mathrm{T}} d_k < 0 \tag{4}$$

which guarantees that d_k is a descent direction of $f(x)$ at x_k. In order to guarantee the global convergence property, we required some times that d_k satisfies the sufficient descent condition:

$$g_k^{\mathrm{T}} d_k \leq -c\|g_k\|^2 \tag{5}$$

where $c>0$ is a constant. Moreover, we need to choose d_k to satisfy the angle property:

$$\cos\langle -g_k, d_k\rangle = -g_k^{\mathrm{T}} d_k \Big/ (\|g_k\|.\|d_k\|) \geq \eta_0$$

(6)

where $\eta_0 \in (0,1]$ is a constant and $\langle -g_k, d_k \rangle$ denotes the angle between the vectors $-g_k$ and d_k.

The commonly used line search rules are as follows see [1, 19, 47, 48]:

(a) **Minimization Rule**: At each iteration, α_k is selected so that:

$$f(x_k + \alpha_k d_k) = \min_{\alpha > 0} f(x_k + \alpha d_k) \tag{7}$$

(b) **Approximate Minimization Rule**: At each iteration , α_k is selected so that:

$$\alpha_k = \min\left\{\alpha \middle| g(x_k + \alpha d_k)^{\mathrm{T}} d_k = 0, \alpha > 0\right\} \tag{8}$$

(c) **Armijo Rule**: Set scalar s_k, β, L, $\sigma > 0$ with:

$$s_k = -g_k^{\mathrm{T}} d_k \Big/ \left(L\|d_k\|^2\right), \beta \in (0,1); \quad \sigma \in (0, 1/2). \tag{9}$$

Let α_k be the larges α in $\{s_k, \beta s_k, \beta^2 s_k, ...\}$ then

$$f_k - f(x_k + \alpha d_k) \geq -\sigma \alpha g_k^{\mathrm{T}} d_k \tag{10}$$

(d) **Limited Minimization Rule**:

if $L > 0$, is a constant; s_k is defined as in (9) then α_k is defined by:

$$f(x_k + \alpha_k d_k) = \min_{\alpha \in [0, s_k]} f(x_k + \alpha d_k) \tag{11}$$

(e) **Goldstein Rule**: if a fixed scalar $\sigma \in (0, 1/2)$ is selected then α_k is chosen to satisfy:

$$\sigma \le \left[f(x_k + \alpha_k d_k) - f_k \right] \Big/ \alpha_k g_k^{\mathrm{T}} d_k \le 1 - \sigma \tag{12}$$

Some important global convergent results for various method using the above mentioned specific line-search procedures have been given in [1] and [31]. In fact, the above mentioned line-search methods are monotone descent for unconstrained optimization see [19] and [22]. Non-monotone line-search methods have been investigated also by many authors see, for example [22] and [32]. Since 1952, there have been many well-known formulas for the scalar β_k, for example, Fletcher-Reeves (FR), Polak-Ribiere-Polyak (PRP), Hestenes-Stiefel (HS):

$$\beta_k^{FR} = \frac{\|g_k\|^2}{\|g_{k-1}\|^2}, \ \beta_k^{PRP} = \frac{g_k^T y_{k-1}}{\|g_{k-1}\|^2}, \ \beta_k^{HS} = \frac{g_k^T y_{k-1}}{d_{k-1}^T y_{k-1}}, \tag{13}$$

where $y_{k-1} = g_k - g_{k-1}$, symbol $\|.\|$ denotes the Euclidean norm of vectors. Their corresponding methods generally specified as FR, PRP, and HS CG-methods. If f is a strictly convex quadratic function, all these methods are equivalent in the case that an exact line search (ELS) is used. If the objective function is non-convex, their behaviors may be distinctly different. In the past two decades, the convergence properties of FR, PRP, and HS methods have been intensively studied by many researchers, see for example [3, 16, 27, 21, 23, 24, 42, 44, 51]. Although the PRP scheme addresses both the jamming of the FR method and the possibility of convergence failure, it interferes with the *n*-step convergence property of the CG method for strongly convex quadratic functions. That is, when the CG-method is applied to a quadratic with an ELS, the successive iterates minimize f over an expanding sequence of subspaces, leading to rapid convergence. In this case, $\beta_k > 0$ for each *k*, however, due to rounding errors we can have $\beta_k^{PRP} < 0$, which implies that $\beta_k^{PRP} = 0$. Each time β_k is set to zero, the CG method is restarted, and the expanding sequence of subspaces reinitiates with a one dimensional space, leading to slower convergence than would be achieved if there was no restart.

Another important issue related to the performance of CG methods is the line search, which requires sufficient accuracy to ensure that the search directions yield descent direction [25]. Common criteria for the line search accuracy are the Wolfe conditions [45, 46]:

$$f(x_{k-1} + \alpha_k d_k) - f(x_{k-1}) \le -\delta \alpha_k g_{k-1}^T d_{k-1}, \tag{14}$$

$$g_k^T d_{k-1} \geq \sigma g_{k-1}^T d_{k-1}, \tag{15}$$

where $0 < \delta \leq \sigma < 1$. In the "strong Wolfe" conditions, (16) is replaced by $\left| g_k^T d_{k-1} \right| \leq -\sigma g_{k-1}^T d_{k-1}$. It has been shown [17] that for the FR scheme, the strong Wolfe conditions may not yield a direction of descent unless $\sigma \leq 1/2$. In typical implementations of the Wolfe conditions, it is often most efficient to choose σ close to one. Hence, the constraint $\sigma \leq 1/2$, needed to ensure descent, represents a significant restriction in the choice of the line search parameters. For the PRP scheme, the strong Wolfe conditions may not yield a direction of descent for any choice of $\sigma \in (0,1)$. However, in practical computation, the HS and PRP methods, which share the common numerator $g_k^T y_{k-1}$, are generally believed to be the most efficient CG methods, and have got meticulous in recent years. One remarkable property of both methods is that they essentially perform a restart if a bad direction occurs [27]. However, Powell [39] constructed an example showed that both methods can cycle infinitely without approaching any stationary point even if an ELS is used. This counter example also indicates that both methods have a drawback that they may not be globally convergent when the objective function is non-convex. Therefore, during the past few years, much effort has been investigated to create new formulae for β_k, which not only possess global convergence for general functions but are also superior to original method from the computation point of view; see [6, 10, 13, 15, 26, 29, 43, 50]. An excellent survey of nonlinear CG methods with special attention to global convergence properties was made by Hager and Zhang [27].

The paper is organized as follows: In Section 2 we present some scaled CG methods. The outlines and the rate of convergence of these new methods for strongly convex functions with Biggs [12]; Al-Bayati's memoryless self-scaling VM updates [4]; Birgin-Martinez [13] and Abbo [2] positive parameters which are used to scale these spectral CG-methods are given in Section 3. In this section also, we have used an acceleration scheme of Andrei's step-size parameter CG method [11]; the idea of this computational scheme is to take an advantage that the step lengths taken in CG methods are different from unity. In Section 4 we have presented some computational results on a set of 100 unconstrained optimization problems from the CUTE [14] collection along with some other large-scale unconstrained optimization problems presented in [8]. Conclusion remarks are pointed out in Section 5. In Section 6 certain open problems have been listed. Finally, in Section 7 details of the test problems are given.

2. Preliminaries

2.1 Scaled CG Methods:

The algorithm generates a sequence x_k of approximations to the minimum x^* of f, in which:

$$x_{k+1} = x_k + \alpha_k d_k, \quad (16)$$

$$d_{k+1} = -\theta_{k+1} g_{k+1} + \beta_k s_k, \quad (17)$$

where α_k is selected to minimize $f(x)$ along the search direction d_k, β_k is a scalar parameter, $s_k = x_{k+1} - x_k$, and θ_{k+1} is a scalar parameter. The iterative process is initialized with an initial point x_0 and $d_0 = -g_0$. Observe that if $\theta_{k+1} = 1$, then we get the classical CG methods according to the value of the scalar parameter β_k. On the other hand, if $\beta_k = 0$, then we get another class of methods according to the selection of the parameter θ_{k+1}. Considering $\beta_k = 0$, there are two possibilities for θ_{k+1}: a positive scalar or a positive definite matrix. If $\theta_{k+1} = 1$, then we have the SD-method. If $\theta_{k+1} = \nabla^2 f(x_{k+1})^{-1}$, or an approximation of it, then we get the Newton or the Quasi-Newton (QN) methods, respectively. Therefore, we see that in the general case, when $\theta_{k+1} \neq 0$ is selected in a QN manner, and for $\beta_k \neq 0$, (17) represents a combination between the QN and the CG methods. However, if θ_{k+1} is a matrix containing some useful information about the inverse Hessian of function f, we are better off using $d_{k+1} = -\theta_{k+1} g_{k+1}$, since the addition of the term $\beta_k s_k$ in (17) may prevent the direction d_k from being a descent direction unless the line search is sufficiently accurate. As we know, when the initial point x_0 is close enough to a local minimum point x^*, then the best direction to be followed in the current point x_{k+1} is the Newton direction $-\nabla^2 f(x_{k+1})^{-1} g_{k+1}$. Therefore, our motivation is to choose the parameter β_k in (17) so that for every $k \geq 1$ the direction d_{k+1} given by (17) can be the best direction we know, i.e. the Newton direction. Hence, using the Newton direction from the equality:

$$-\nabla^2 f(x_{k+1})^{-1} g_{k+1} = -\theta_{k+1} g_{k+1} + \beta_k s_k \quad (18)$$

yields:

$$\beta_k = \frac{-s_k^T \nabla^2 f(x_{k+1}) \theta_{k+1} g_{k+1} - s_k^T g_{k+1}}{s_k^T \nabla^2 f(x_{k+1}) s_k} \tag{19}$$

Observing that if the line search is exact we get the scaled Daniel method [18]. For large-scale problems, choices for the update parameter that do not require the evaluation of the Hessian matrix are often preferred in practice to the methods that require the Hessian. Now, for QN methods an approximation matrix β_k to the Hessian $\nabla^2 f(x_k)$ is used and updated so that the new matrix B_{k+1} satisfies the secant condition $B_{k+1} s_k = y_k$. Therefore, in order to have an algorithm for solving large-scale problems we can assume that the pair (s_k, y_k) satisfies the secant condition. In this case, Zhang et al. [49] proved that if $\|s_k\|$ is sufficiently small, then:

$$\beta_k = \frac{(\theta_{k+1} y_k - s_k)^T g_{k+1}}{y_k^T s_k} \tag{20}$$

Birgin and Martínez [13] arrived at the same formula for β_k, but using a geometric interpretation of quadratic function minimization. The direction corresponding to β_k given in (20) is as follows:

$$d_k = -\theta_{k+1} g_{k+1} + \frac{(\theta_{k+1} y_k - s_k)^T g_{k+1}}{y_k^T s_k} s_k \tag{21}$$

The following particularizations are obvious. If $\theta_{k+1} = 1$, then (21) is the direction considered by Perry [34]. At the same time we see that (21) is the direction given by Dai and Liao [15] for *t* = 1, obtained this time by an interpretation of the conjugcy condition. Additionally, if $s_j^T g_{j+1} = 0, \quad j = 0,1,...,k$ then from (21) we get:

$$d_{k+1} = -\theta_{k+1} g_{k+1} + \frac{\theta_{k+1} y_k^T g_{k+1}}{\alpha_k \theta_{k+1} g_k^T g_k} s_k \tag{22}$$

which is the direction corresponding to a generalization of the PR formula. Of course, if $\theta_{k+1} = \alpha_k = 1$ in (22), we get the classical PRP formula [35, 36]. If $s_j^T g_{j+1} = 0, \quad j = 0,1,...,k$ and additionally the successive gradients are orthogonal, then from (21) we get:

$$d_{k+1} = -\theta_{k+1} g_{k+1} + \frac{\theta_{k+1} g_{k+1}^T g_{k+1}}{\alpha_k \theta_{k+1} g_k^T g_k} s_k \tag{23}$$

which is the direction corresponding to a generalization of the FR formula [20]. Therefore, (21) is a general formula for direction computation in a CG manner including the classical FR [20] and PRP [35, 36] formulae.

2.2 New Memoryless VM-Methods:

The extension to the scaled CG is very simple. Using the same methodology as considered by Shanno [40] , we get the following new direction d_{k+1} in our new three-term Memoryless VM method:

$$d_{k+1} = -\theta_{k+1} g_{k+1} + \theta_{k+1} \left(\frac{g_{k+1}^T s_k}{y_k^T s_k} \right) y_k - \left[\left(\rho_k + \theta_{k+1} \frac{y_k^T y_k}{y_k^T s_k} \right) \frac{g_{k+1}^T s_k}{y_k^T s_k} - \theta_{k+1} \frac{g_{k+1}^T y_k}{y_k^T s_k} \right] s_k \tag{24a}$$

where

$$\rho_k = \frac{y_k^T y_k}{y_k^T s_k} \qquad \text{(Al-Bayati [4])} \tag{24b}$$

$$\rho_k = \frac{6}{s_k^T y_k} \left[f(x_k) - f(x_{k+1}) + s_k^T g_{k+1} \right] - 2 \qquad \text{(Biggs [12])} \tag{24c}$$

and

$$\theta_{k+1} = \frac{s_k^T s_k}{y_k^T s_k} \qquad \text{(Birgin-Martinez [13])} \tag{24d}$$

$$\theta_{k+1} = \frac{s_k^T s_k}{s_k^T s_k + \gamma\, s_k{}^T y_k} \qquad \text{(Abbo [2])}$$

(24e)

involving only four scalar products. Let us call these methods as:

Equation (24) with ρ_k = Al-Bayati + θ_{k+1} =Abbo (call **New1**)
Equation (24) with ρ_k = Al-Bayati + θ_{k+1} = Birgin-Martinez (call **New2**)
Equation (24) with ρ_k = Biggs + θ_{k+1} = Birgin-Martinez (call **New3**)
Equation (24) with ρ_k = Biggs + θ_{k+1} = Abbo (call **New4**)

Again observe that if $g_{k+1}^T s_k = 0$, then (24) reduces to:

$$d_{k+1} = -\theta_{k+1} g_{k+1} + \theta_{k+1} \frac{g_{k+1}^T y_k}{y_k^T s_k} s_k$$

(25)

Thus, in this case, the effect is simply one of multiplying the HS [28] search direction by a positive scalar. In order to ensure the convergence of the new method with d_{k+1} given by (24), we need to constrain the choice of α_k. We consider line searches that satisfy the Wolfe conditions [45, 46] given in (14) and (15).

Theorem 2.3 Suppose that α_k in (16) satisfies the Wolfe conditions (14) and (15), then the directions d_{k+1} given by (24) with ELS are descent directions.

Proof. Since $d_0 = -g_0$, we have $g_0^T d_0 = -\|g_0\|^2 \le 0$. By ELS (24) reduces to (25). Multiplying (25) by g_{k+1}^T, we have:

$$g_{k+1}^T d_{k+1} = \frac{1}{\left(y_k^T s_k\right)^2}\left(-\theta_{k+1}\|g_{k+1}\|^2 \left(y_k^T s_k\right)^2 + 2\theta_{k+1}\left(g_{k+1}^T y_k\right)\left(g_{k+1}^T s_k\right)\left(y_k^T s_k\right)\right.$$
$$\left. -\left(g_{k+1}^T s_k\right)^2 \left(y_k^T s_k\right) - \theta_{k+1}\left(y_k^T y_k\right)\left(g_{k+1}^T s_k\right)^2\right)$$

Applying the inequality $u^T v \le \frac{1}{2}\left(\|u\|^2 + \|v\|^2\right)$ to the second term of the right hand side of the above equality, with $u = \left(s_k^T y_k\right) g_{k+1}$ and $v = \left(g_{k+1}^T s_k\right) y_k$ we get:

$$g_{k+1}^T d_{k+1} \le -\frac{\left(g_{k+1}^T s_k\right)^2}{y_k^T s_k}$$

(26)

But, by Wolfe condition (15), $y_k^T s_k > 0$. Therefore, $g_{k+1}^T d_{k+1} < 0$, for every $k = 0,1,...$. We have observed that the second Wolfe condition (15) is crucial for the descent character of direction (25). Besides, we see that the estimation (26) is independent of the parameter θ_{k+1}. Usually, all CG algorithms are periodically restarted. The Powell restarting procedure [37, 38] is to test if there is very little orthogonality left between the current gradient and the previous one. At step k when:

$$\left|g_{k+1}^T g_k\right| \ge 0.2\left\|g_{k+1}\right\|^2$$

(27)

we restart the algorithm to the SD-direction.

3. Step-Size Accelerations:

In the CG methods the search directions tend to be poorly scaled and as a consequence the line search must perform more function evaluations in order to obtain a suitable step-length α_k. In order to improve the performances of the CG methods the efforts were directed to design procedures for direction computation based on the second order information. Jorge Nocedal [33] pointed out that in CG methods the step lengths may differ from unity in a very unpredictable manner. They can be larger or smaller than unity depending on how the problem is scaled. Numerical comparisons between CG methods and the limited memory QN method, by Liu and Nocedal [30], show that the latter is more successful [5]. Here, we have pointed out Andrei's [11] acceleration scheme; basically, this modifies the step length in a multiplicative manner to improve the reduction of the function values along the iterations [7, 9].

3.1 Outline of the New Proposed Algorithms.

Having in view the above developments and the definitions of g_k, s_k and y_k, as well as the selection procedure for θ_{k+1} computation, the out lines of the following proposed new algorithm can be presented as follows:

Step 1. Initialization. Select $x_0 \in R^n$, and the parameters $\gamma = .5,\ 0 < \delta \le \sigma < 1.$ Compute $f(x_0)$ and $g_0 = \nabla f(x_0)$. Set $d_0 = -g_0$ and $\alpha_0 = 1/\|g_0\|$. Set $k = 0$.

Step 2. Acceleration scheme of Andrei's [11] line search:
Compute α_k satisfying Wolfe's conditions (14) and (15).
Update the variables $x_{k+1} = x_k + \alpha_k d_k$.
Compute $f(x_{k+1}),\ g_{k+1}$ and set $a_k = g_k^T d_k;\ \ b_k = (g_k - g_{k+1})^T d_k$.
If $b_k \neq 0$, then set $\gamma_k = a_k / b_k$ and update the variables as $x_{k+1} = x_k + \gamma_k \alpha_k d_k$.
Compute $f(x_{k+1}),\ g_{k+1},\ s_k$ and y_k.
Otherwise, if $b_k = 0$, then compute $s_k,\ y_k$.

Step 3. Test for convergence, i.e. if $(\ \|g_k\|_\infty < \max\ (\ 10^{-6},\ 10^{-10}\|g_0\|_\infty\ )\ )$ is satisfied then the iterations are stopped.

Step 4. If Powell restarting criterion (27) is satisfied, then (Do a restart step by a SD-direction); otherwise continue.

Step 5. Compute the new three-term search direction d_k as in (24), with saved and
different values of the parameters $\theta,\ \rho,\ s$ and y.

Step 6. Set $k = k+1$ and go to **Step 2**.

3.2 Convergence Analysis for Strongly Convex Functions:

Throughout this section we assume that f is strongly convex and ∇f is Lipschitz continuous on the level set:

$$S = \{x \in R^n : f(x) \le f(x_0)\} \tag{28}$$

That is, there exists constants $\mu > 0$ and L such that

$$(\nabla f(x)-\nabla f(y))^T(x-y)\geq \mu\|x-y\|^2$$
(29)

and

$$\|\nabla f(x)-\nabla f(y)\|\geq L\,\|x-y\|$$
(30)

for all x and y from S. For the convenience we include here the following lemma [26].

Lemma 3.3. Assume that d_k is a descent direction and ∇f satisfies the Lipschitz condition

$$\|\nabla f(x)-\nabla f(x_k)\|\geq L\,\|x-x_k\|$$
(31)

for every x on the line segment connecting x_k and x_{k+1}, where L is a constant. If the line search satisfies the second Wolfe condition (15), then

$$\alpha_k\geq\frac{1-\sigma}{L}\frac{|g_k^T d_k|}{\|d_k\|^2}$$
(32)

Proof.

Subtracting $g_k^T d_k$ from both sides of (15) and using the Lipschitz condition we have

$$(\sigma-1)g_k^T d_k\leq(g_{k+1}-g_k)^T d_k\leq L\,\alpha_k\,\|d_k\|^2$$
(33)

Since d_k is a descent direction and $\sigma<1$, (32) follows immediately from (33). Therefore, satisfying the Wolfe line search conditions α is bounded away from zero, i.e. there exists a positive constant ω, such that $\alpha\geq\omega$.

Lemma 3.4. Assume that f is strongly convex and ∇f is Lipschitz continuous on S. If θ_{k+1} is selected by either (24d) or (24e), then the direction d_{k+1} given by (24) satisfies:

$$\|d_{k+1}\| \leq \left(\frac{2}{\mu} + \frac{2L}{\mu^2} + \frac{L^2}{\mu^3} \right) \|g_{k+1}\|$$

(34a)

Or

$$\|d_{k+1}\| \leq \left(\frac{2}{1+\gamma\mu} + \frac{2L}{(1+\gamma\mu)^2} + \frac{L^2}{(1+\gamma\mu)^3} \right) \|g_{k+1}\|$$

(34b)

Where γ and μ are small positive scalars

Proof.

By Lipschitz continuity (30) we have

$$\|y_k\| = \|g_{k+1} - g_k\| = \|\nabla f(x_k + \alpha_k d_k) - \nabla f(x_k)\| \leq L\, \alpha_k \|d_k\|$$
$$= L\|s_k\|$$

(35)

On the other hand, by strong convexity (29)

$$y_k^T s_k \geq \mu \|s_k\|^2$$

(36)

Selecting θ_{k+1} as in (24d), it follows that

$$\theta_{k+1} = \frac{s_k^T s_k}{y_k^T s_k} \leq \frac{\|s_k\|^2}{\mu \|s_k\|^2} = \frac{1}{\mu}$$

(37a)

Selecting θ_{k+1} as in (24e), it follows that

$$\theta_{k+1} = \frac{s_k^T s_k}{s_k^T s_k + \gamma\, s_k^T y_k} \leq \frac{\|s_k\|^2}{\|s_k\|^2 + \gamma\mu \|s_k\|^2} = \frac{1}{1+\gamma\mu}$$

(37b)

Now, using the triangle inequality and the above estimates (35)–(37), after some algebra on $\|d_{k+1}\|$, where d_{k+1} is given by (24), we get (34). The convergence of the scaled CG method when f is strongly convex is given by:

Theorem 3.5. Assume that f is strongly convex and ∇f is Lipschitz continuous on the level set S. If at every step of the CG direction given in (16) with d_{k+1} given by (24) and the step length α_k selected to satisfy the Wolfe conditions (14) and (15), then either $g_k = 0$ for some k, or $\lim_{k\to\infty} g_k = 0$.

Proof.

Suppose $g_k \neq 0$ for all k. By strong convexity we have

$$y_k^{\mathrm{T}} d_k = (g_{k+1} - g_k)^T d_k \geq \mu \alpha_k \|d_k\|^2 \quad (38)$$

By **Theorem 2.3**, $g_k^T d_k < 0$. Therefore, the assumption $g_k \neq 0$ implies $d_k \neq 0$. Since $\alpha_k > 0$, from (38) it follows that $y_k^T d_k > 0$. But f is strongly convex over S, therefore, f is bounded from below. Now, summing over k the first Wolfe condition (14) we have:

$$\sum_{k=0}^{\infty} \alpha_k g_k^T d_k > -\infty$$

Considering the lower bound for α_k given by (32) in **Lemma 3.3** and having in view that d_k is a descent direction it follows that:

$$\sum_{k=1}^{\infty} \frac{\left|g_k^T d_k\right|^2}{\|d_k\|^2} < \infty \quad (39)$$

Now, from (26), using the inequality of Cauchy and (36) we get for (24d) and (24e) respectively:

$$g_{k+1}^T d_{k+1} \le -\frac{\left(g_{k+1}^T s_k\right)^2}{y_k^T s_k} \le -\frac{\left\|g_{k+1}\right\|^2 \left\|s_k\right\|^2}{\mu \left\|s_k\right\|^2} = -\frac{\left\|g_{k+1}\right\|^2}{\mu}$$

or

$$g_{k+1}^T d_{k+1} \le -\frac{\left(g_{k+1}^T s_k\right)^2}{s_k^T s_k + \gamma y_k^T s_k} \le -\frac{\left\|g_{k+1}\right\|^2 \left\|s_k\right\|^2}{\left\|s_k\right\|^2 + \mu \left\|s_k\right\|^2} = -\frac{\left\|g_{k+1}\right\|^2}{1+\gamma\mu}$$

Therefore, from (39) it follows that

$$\sum_{k=0}^{\infty} \frac{\left|g_k\right|^4}{\left\|d_k\right\|^2} < \infty \tag{40}$$

Now, inserting the upper bound (34), for d_k in (40) yields

$$\sum_{k=0}^{\infty} \left\|g_k\right\|^2 < \infty$$

which completes the proof.

4. Numerical Results:

The main work of this section is to report the performance of the new proposed scaled memoryless VM methods on a set of (50) test problems. The codes are written in FORTRAN 77 and in double precision arithmetic. All the tests are performed on a PC. Our experiments are performed on a set of (50) nonlinear unconstrained problems that have second derivatives available. These test problems are contributed in [14] and [8] and their details are given in the Appendix. For each test function we have considered 2 numerical experiments with number of variables *n* = 100 *and* 1000. In order to assess the reliability of our new proposed methods, we have tested it against the standard scaled memoryless BFGS method using the same test problems. All these methods terminate when the following stopping criterion is met:

$$\text{If} \quad \left(\left\|g_k\right\|_\infty < \max \ \left(10^{-6}, 10^{-10} \left\|g_0\right\|_\infty \right) \right) \tag{41}$$

We also force these routines stopped if the iterations exceed 1000 or the number of function evaluations reach 2000 without achieving convergence. We use $\delta=10^{-4}$, $\sigma=0.1$ in the line search routine (14) and (15). **Table 4.1** compares some numerical results for the new methods (namely; **New1** and **New2**) against the scaled memoryless BFGS method; this table indicates for (n) as a dimension of the problem; (NOI), number of iterations; (NOFG), number of function and gradient evaluations; (TIME), the total time required to complete the evaluation process for each test problem. **Table 4.2** compares some numerical results for the new methods (namely; **New3** and **New4**) against the scaled memoryless BFGS method. In **Tables 4.3** and **4.4** we have compared the percentage performance of the new methods (**New1**, **New2**, **New3** and **New4**) against the standard scaled memoryless BFGS method taking over all the tools as 100%.

Table (4.1). Comparison between New1; New2 against scaled memoryless BFGS method for the total of (50) different test problems with dimensions n= 100 and 1000

Prob.	New1 Method NOI/NOFG/TIME	New2 Method NOI/NOFG/TIME	Scaled BFGS Method NOI/NOFG/TIME
1	90/126/0.02	92/126/0.00	23/45/0.00
2	164/223/0.12	138/180/0.17	39/78/0.03
3	176/246/0.02	198/275/0.02	55/103/0.00
4	93/170/0.01	95/185/0.01	47/93/0.02
5	38/74/0.02	64/111/0.02	23/43/0.00
6	22/46/0.00	33/61/0.00	8/24/0.00
7	273/329/0.09	252/304/0.10	469/603/0.06
8	376/411/0.20	373/412/0.25	319/448/0.08
9	13/26/0.00	13/26/0.00	8/18/0.02
10	248/306/0.11	241/285/0.08	240/336/0.06
11	96/188/0.06	99/355/0.07	270/6557/1.41
12	59/96/0.02	54/96/0.02	44/93/0.00
13	32/76/0.00	37/69/0.00	18/42/0.00

14	43/76/0.02	25/59/0.02	15/28/0.02
15	139/191/0.03	234/276/0.02	112/175/0.01
16	10/28/0.00	10/57/0.00	7/14/0.00
17	18/48/0.01	18/48/0.02	7/16/0.00
18	63/108/0.02	78/130/0.00	14/30/0.00
19	32/80/0.02	41/70/0.01	13/28/0.00
20	39/93/0.01	48/137/0.02	43/80/0.00
21	416/634/0.05	405/757/0.05	100/204/0.01
22	12/16/0.00	12/16/0.00	24/26/0.02
23	178/214/0.04	181/216/0.03	831/1147/0.09
24	274/315/0.05	270/325/0.04	406/527/0.09
25	36/153/0.02	36/69/0.00	16/41/0.00
26	145/293/0.12	255/432/0.24	44/99/0.01
27	8/52/0.00	10/56/0.00	4/10/0.00
28	66/101/0.00	69/112/0.02	66/102/0.01
29	21/69/0.00	31/79/0.00	19/133/0.00
30	26/87/0.00	26/86/0.00	20/37/0.00
31	32/91/0.00	38/112/0.02	12/26/0.00
32	21/53/0.00	22/61/0.00	12/24/0.00
33	21/62/0.00	22/64/0.00	21/37/0.02
34	24/59/0.00	25/62/0.00	27/49/0.02
35	196/230/0.08	204/236/0.06	283/386/0.15
36	245/301/2.08	230/276/1.84	358/563/2.27
37	141/186/0.03	203/247/0.01	66/117/0.00
38	275/332/0.03	257/298/0.05	402/530/0.05

39	291/352/0.05	289/350/0.05	481/618/0.08
40	54/93/0.00	61/99/0.01	50/98/0.01
41	342/399/0.04	242/315/0.02	27/50/0.00
42	13/26/0.00	13/26/0.00	8/18/0.00
43	329/390/0.11	350/412/0.11	281/384/0.06
44	234/261/0.07	223/252/0.06	307/409/0.06
45	258/292/0.09	230/264/0.08	280/369/0.07
46	297/363/0.09	346/403/0.11	307/428/0.06
47	212/243/0.07	223/270/0.06	272/371/0.07
48	73/119/0.01	80/130/0.00	125/2426/0.12
49	35/89/0.02	32/89/0.02	27/124/0.02
50	29/63/0.00	25/70/0.01	11/24/0.00
Total	6328/8879/3.83	6553/9446/3.72	6661/18231/5

Table (4.2). Comparison between New3; New4 against scaled memoryless BFGS method for the total of (50) different test problems with dimensions n= 100 and 1000

Prob.	New3 Method NOI/NOFG/TIME	New4 Method NOI/NOFG/TIME	Scaled BFGS Method NOI/NOFG/TIME
1	27/63/0.01	29/75/0.00	23/45/0.00
2	111/155/0.13	79/131/0.03	39/78/0.03
3	74/182/0.01	80/204/0.00	55/103/0.00
4	70/154/0.00	70/163/0.02	47/93/0.02
5	39/86/0.01	34/68/0.00	23/43/0.00
6	14/35/0.00	13/34/0.00	8/24/0.00
7	285/324/0.09	290/335/0.11	469/603/0.06
8	346/377/0.11	337/368/0.11	319/448/0.08

9	13/35/0.00	13/26/0.00	8/18/0.02
10	236/289/0.10	262/311/0.10	240/336/0.06
11	132/1668/0.34	183/3264/0.67	270/6557/1.41
12	46/95/0.02	45/94/0.01	44/93/0.00
13	31/62/0.00	24/63/0.00	18/42/0.00
14	25/64/0.02	26/67/0.04	15/28/0.02
15	74/111/0.01	86/129/0.01	112/175/0.01
16	7/30/0.00	10/55/0.00	7/14/0.00
17	18/48/0.01	15/48/0.02	7/16/0.00
18	29/86/0.00	23/49/0.00	14/30/0.00
19	31/77/0.02	33/68/0.02	13/28/0.00
20	32/98/0.01	36/96/0.00	43/80/0.00
21	133/352/0.02	120/295/0.00	100/204/0.01
22	12/16/0.01	12/16/0.00	24/26/0.02
23	214/257/0.03	181/225/0.03	831/1147/0.09
24	304/348/0.07	303/356/0.04	406/527/0.09
25	96/2320/0.15	25/65/0.00	16/41/0.00
26	78/237/0.10	78/190/0.05	44/99/0.01
27	10/56/0.00	8/52/0.00	4/10/0.00
28	65/104/0.02	66/106/0.02	66/102/0.01
29	23/69/0.00	26/77/0.00	19/133/0.00
30	25/68/0.00	25/75/0.00	20/37/0.00
31	27/84/0.02	23/68/0.00	12/26/0.00
32	16/39/0.01	20/54/0.00	12/24/0.00
33	21/61/0.02	21/63/0.00	21/37/0.02
34	24/60/0.02	24/60/0.02	27/49/0.02
35	233/267/0.11	240/275/0.08	283/386/0.15

36	254/307/2.14	267/331/2.25	358/563/2.27
37	78/128/0.02	82/144/0.01	66/117/0.00
38	315/365/0.04	271/337/0.05	402/530/0.05
39	263/317/0.05	264/306/0.05	481/618/0.08
40	50/146/0.02	60/108/0.00	50/98/0.01
41	41/80/0.01	48/97/0.00	27/50/0.00
42	15/45/0.00	15/45/0.02	8/18/0.00
43	527/557/0.22	231/265/0.06	281/384/0.06
44	490/515/0.16	242/276/0.08	307/409/0.06
45	250/285/0.11	230/264/0.08	280/369/0.07
46	522/552/0.19	234/270/0.08	307/428/0.06
47	296/322/0.09	222/262/0.09	272/371/0.07
48	57/203/0.02	55/238/0.01	125/2426/0.12
49	27/89/0.01	28/92/0.02	27/124/0.02
50	19/58/0.02	16/57/0.00	11/24/0.00
Total	6125/12346/4.57	5125/10717/4.18	6661/18231/5

Table 4.3. Percentage performance of the New1, New2 methods against the scaled memoryless BFGS method for the total of (50) test problems.

Tools	Scaled BFGS	New1	New2
NOI	100%	95%	98%
NOFG	100%	49%	52%
TIME	100%	77%	74%

Table 4.4. Percentage performance of the New3, New4 methods against the scaled memoryless BFGS method for the total of (50) test problems.

Tools	Scaled BFGS	New3	New4
NOI	100%	92%	77%

NOFG	100%	68%	59%
TIME	100%	91%	84%

From **Tables (4.3) and (4.4)** we have obtained the following results:
New1 saves **05%** NOI; **51%** NOFG and **23%** TIME, compared with **scaled BFGS**.
New2 saves **02%** NOI; **48%** NOFG and **26%** TIME, compared with **scaled BFGS**.
New3 saves **08%** NOI; **32%** NOFG and **09%** TIME, compared with **scaled BFGS**.
New4 saves **23%** NOI; **41%** NOFG and **16%** TIME, compared with **scaled BFGS**.

5. Conclusions:

We have presented four new three-term CG methods which they are assumed to be an accelerations scheme of Al-Bayati's [4] and Biggs's [12] VM updates. The acceleration scheme is simple and proved to be robust in numerical experiments. For general functions the convergence of the methods is coming from **Theorem 3.5** and the restart procedure. Therefore, for strongly convex functions and under inexact line searches (ILS) the methods are very close to the Shanno computational scheme [40, 41] which are the scaled memoryless BFGS method where the scaling factor is the inverse of a scalar approximation of the Hessian. If the Powell restart criterion (27) is used, for general functions f bounded from below with bounded second partial derivatives and bounded level set, using the same arguments considered by Shanno in [40] it is possible to prove that the iterates converge to a point x^*. Under certain conditions we have proved that the new methods are globally convergent. For uniformly convex functions the reduction in the function values is significantly improved for a set of 100 test unconstrained optimization problems with dimensions 100 and 1000 variables.

6. Open Problems:

1- The new proposed methods, given in (24), may be implemented by an initial scaling parameter for example, $\theta_{k+1} = \alpha_0 \theta_{k+1}$. Shanno [40] proved that the CG methods are exactly the BFGS VM-method, where at every step the approximation to the inverse Hessian is restarted as the identity matrix. Now we extend this result for the new method defined in (24) by:

$$d_+ = -\left\{\alpha_0\theta I - \alpha_0\theta\frac{ys^T + sy^T}{y^T s} + \left[\rho + \alpha_0\theta\frac{y^T y}{y^T s}\right]\frac{ss^T}{y^T s}\right\}g_+$$

(42)

Thus, the effect is simply multiplying the search direction by a positive initial scaling parameter α_0; d_+, g_+ denotes the values of the next direction and next gradient respectively.

2- Another new estimation for the parameter ρ_k may be implemented in our new proposed methods (24) as follows:

$$\gamma_k = \frac{1}{g_k{}^T g_k}\left[f(x_k) - f(x_{k+1}) - (0.5)g_k^T g_k\right] + \delta, \quad \delta > 0$$

(43)

If $\gamma_k \geq 0$; set $\rho_k = \gamma_k$; otherwise set

$$\rho_k = \frac{2}{g_k{}^T g_k}(1/\gamma_k^2)\left[f(x_{k+1}) - f(x_k) + (0.5 + \gamma_k)g_k^T g_k\right]$$

(44)

7. Appendix.

The details of the test functions, used in this paper, can be found in [14]. The numbers (1-50) in our tables indicate to:

(1)-Extended Freudenstein & Roth Function.
(2)-Extended Trigonometric Function.
(3)-Extended Rosenbrock Function
(4)-Extended White & Holst function
(5)-Extended Beale Function U63 (MatrixRom) Function.
(6)-Extended Penalty Function.
(7)-Perturbed Quadratic function.
(8)-Raydan 1 Function.
(9)-Raydan 2 Function.
(10)-Diagonal2 Function.
(11)-Hager Function.
(12)-Generalized Tridiagonal-1 Function.
(13)-Extended Tridiagonal-1 Function.
(14)-Extended 3-Exp. Terms Function.
(15)-Generalized Tridiagonal-2 Function.
(16)-Diagonal4 Function.

(17)-Diagonal5 Function.
(18)-Extended Himmelblau Function.
(19)-Extended PSC1 Function.
(20)-Extended Block Diag. BD1 Function.
(21)-Extended Maratos Function.
(22)-Extended Cliff Function.
(23)-Quadratic Diagonal Perturbed Function.
(24)-Quadratic Function QF1 Function.
(25)-Extended Quadratic Penalty QP1 Function.
(26)-Extended Quadratic Penalty QP2 Function.
(27)-Extended EP1 Function.
(28)-Extended Tri-diagonal 2 Function.
(29)-ARWHEAD (CUTE)-Function.
(30)-NONDIA (Shanno-78) (CUTE) Function.
(31)-DQDRTIC Function.
(32)-DIXMAANA (CUTE)-Function.
(33)-DIXMAANB (CUTE)-Function.
(34)-DIXMAANC (CUTE)-Function.
(35)-DIXMAANE (CUTE) Function.
(36)-Partial Perturbed Quadratic Function.
(37)-Broyden Tridiagonal Function.
(38)-Almost Perturbed Quadratic Function.
(39)-Tridiagonal Perturbed Quadratic Function.
(40)-EDENSCH (CUTE)-Function.
(41)-LIARWHD (CUTE) Function.
(42)-DIAGONAL 6 Function.
(43)-DIXMAANF (CUTE) Function.
(44)-DIXMAANG (CUTE) Function.
(45)-DIXMAANI (CUTE) Function.
(46)-DIXMAANJ (CUTE) Function.
(47)-DIXMAANK (CUTE) Function.
(48)-ENGVAL1 (CUTE) Function.
(49)-COSINE (CUTE) Function.
(50)-DENSCHNB (CUTE) Function.

References:

[1] L. Armijo, Minimization of function having Lipschitz continuous first partial derivatives , pacific. J. Math., vol. 16 (1966) 1-13.

[2] KH. K. Abbo, Modifying of Barzilai and Borwein method for solving large-scale unconstrained optimization problems, Iraqi J. of Statistical Science, Mosul, Iraq, vol. 7(11) (2007) 27-64.

[3] M. Al-Baali, Descent property and global convergence of the Fletcher-Reeves method with inexact line search, IMA Journal of Numerical Analysis, vol. 5(1) (1985) 121–124.

[4] A. Y. Al- Bayati, (A new family of self-scaling VM algorithms for unconstrained optimization) J. Educ. and Sci, Mosul, Iraq, vol.12 (1991) 25-54.

[5] N. Andrei, Numerical comparison of conjugate gradient algorithms for unconstrained optimization, Studies in Informatics and Control, vol. 16 (2007) 333–352.

[6] N. Andrei, "Scaled conjugate gradient algorithms for unconstrained optimization," Computational Optimization and Applications, vol. 38(3) (2007) 401–416.

[7] N. Andrei, Accelerated hybrid conjugate gradient algorithm with modified secant condition for unconstrained optimization, ICI Technical Report, (2008).

[8] N. Andrei, An unconstrained optimization test functions collection, Advanced Modeling and Optimization. An Electronic International Journal, vol. 10 (2008) 147–161.

[9] N. Andrei, Acceleration of conjugate gradient algorithms for unconstrained optimization, Applied Mathematics and Computation, vol. 213 (2009) 361–369.

[10] N. Andrei, "Another nonlinear conjugate gradient algorithm for unconstrained optimization," Optimization Methods & Software, vol. 24(1) (2009) 89–104.

[11] N. Andrei, Accelerated scaled memoryless BFGS preconditioned conjugate gradient algorithm for unconstrained optimization European Journal of Operational Research, vol. 204 (2010) 410–420.

[12] M. C. Biggs, A note of minimization algorithm which make use of non-Quadratic properties of the objective function, Journal of Institute of Mathematics and its Application, vol. 12 (1973) 337-338.

[13] E. G. Birgin and J. M. Martinez, A spectral conjugate gradient method for unconstrained optimization, Applied Mathematics and Optimization, vol. (43) (2001), 117-128.

[14] I. Bongartz; A. R. Conn; N. I. M. Gould and P. L. Toint, CUTE: Constrained and unconstrained testing environments, ACM Transactions on Mathematical Software, vol. 21 (1995) 123–160.

[15] Y. H. Dai and L. Z. Liao, "New conjugacy conditions and related nonlinear conjugate gradient methods," Applied Mathematics and Optimization, vol. 43(1) (2001) 87–101.

[16] Y. H. Dai and Y. Yuan, Convergence properties of the Fletcher-Reeves method, IMA Journal of Numerical Analysis, vol. 16(2) (1996) 155–164.

[17] Y. H. Dai, and Y. Yuan, "Nonlinear Conjugate Gradient Methods," Shang Hai Science and Technology, Beijing, (2000).

[18] J. W. Daniel, The conjugate gradient method for linear and nonlinear operator equations, SIAM Journal on Numerical Analysis, vol. 4 (1967) 10–26.

[19] A. V. Fiacco and G .P. McCormick, Nonlinear programming , Sequential Unconstrained Minimization Techniques, SIAM, Philadelphia, (1990).

[20] R. Fletcher and C. M. Reeves, Function minimization by conjugate gradients, Computer Journal, vol. 7 (1964) 149–154.

[21] J. C. Gilbert and J. Nocedal, "Global convergence properties of conjugate gradient methods for optimization," SIAM Journal on Optimization, vol. 2(1) (1992) 21–42.

[22] A. A. Goldstein and J. F. Price, An effective algorithm for minimization, Numer. Math., vol. 10 (1967) 184-189.

[23] L. Grippo and S. Lucidi, "A globally convergent version of the Polak-Ribiere conjugate gradient method," Mathematical Programming, vol. 78(3) (1997) 375–391.

[24] L. Grippo and S. Lucidi, "Convergence conditions, line search algorithms and trust region implementations for the Polak Ribiere conjugate gradient method," Optimization Methods & Software, vol. 20(1) (2005) 71–98.

[25] W. W. Hager, "A derivative-based bracketing scheme for univariate minimization and the conjugate gradient method. Comput. Math. Appl., vol. 18 (1989) 779–795.

[26] W. W. Hager and H. Zhang, "A new conjugate gradient method with guaranteed descent and an efficient line search," SIAM Journal on Optimization, vol. 16(1) (2005) 170–192.

[27] W. W. Hager and H. Zhang, "A survey of nonlinear conjugate gradient methods," Pacific Journal of Optimization, vol. 2 (2006) 35–58.

[28] M. R. Hestenes and E. Stiefel, Methods of conjugate gradients for solving linear systems, Journal of Research of the National Bureau of Standards Section B, vol. 48 (1952) 409–436.

[29] G. Li; C. Tang, and Z. Wei, "New conjugacy condition and related new conjugate gradient methods for unconstrained optimization," Journal of Computational and Applied Mathematics, vol. 202(2) (2007) 523–539.

[30] D. C. Liu and J. Nocedal, On the limited memory BFGS method for large scale optimization methods, Mathematical Programming, vol. 45 (1989) 503–528.

[31] J. J. More; B. S. Garbow and K. E. Hillstrom, Testing unconstrained optimization software, ACM Trans, Math. Software, vol. 7 (1981) 17-41.

[32] J. Nocedal, Theory of Algorithms for Unconstrained Optimization , Acta. Numer., vol. 1 (1992) 199-242.

[33] J. Nocedal, Conjugate gradient methods and nonlinear optimization, in: L. Adams, J. L. Nazareth (Eds.), Linear and Nonlinear Conjugate Gradient Related Methods, SIAM, Philadelphia, PA, (1996), 9-23.

[34] J. M. Perry, A class of conjugate gradient algorithms with a two step variable metric memory, Discussion Paper 269, Center for Mathematical Studies in Economics and Management Science, Northwestern University, (1977).

[35] E. Polak and G. Ribiere, Note sur la convergence de methods de directions conjugres, Revue Francaise Informat Reserche Opérationnelle, vol. 16 (1969) 35–43.

[36] B. T. Polyak, The conjugate gradient method in extreme problems, USSR Computational Mathematics and Mathematical Physics, vol. 9 (1969) 94–112.

[37] M. J. D. Powell, Some convergence properties of the conjugate gradient method, Mathematical Programming, vol. 11 (1976) 42–49.

[38] M. J. D. Powell, Restart procedures for the conjugate gradient method, Mathematical Programming, vol. 12 (1977) 241–254.

[39] M. J. D. Powell, "Non-convex minimization calculations and the conjugate gradient method," in Numerical Analysis (Dundee, 1983), vol. 1066 (1984) 122-141.

[40] D. F. Shanno, Conjugate gradient methods with inexact searches, Mathematics of Operations Research, vol. 3 (1978) 244–256.

[41] D. F. Shanno, On the convergence of a new conjugate gradient algorithm, SIAM Journal on Numerical Analysis, vol. 15 (1978) 1247–1257.

[42] J. Sun and J. Zhang, "Global convergence of conjugate gradient methods without line search," Annals of Operations Research, vol.103 (2001)161–173.

[43] Z. Wei; G. Li, and L. Qi, "New quasi-Newton methods for unconstrained optimization problems," Applied Mathematics and Computation, vol. 175(2) (2006) 1156–1188.

[44] Z. Wei; G. Y. Li, and L. Qi, "Global convergence of the Polak- Ribiere - Polyak conjugate gradient method with an Armijo-type inexact line search for non-convex unconstrained optimization problems," Mathematics of Computation, vol. 77(264) (2008) 2173–2193.

[45] P. Wolfe, Convergence conditions for ascent methods, SIAM Review, vol. 11 (1969) 226–235.

[46] P. Wolfe, Convergence conditions for ascent methods II: Some corrections, SIAM Review, vol. 13 (1971) 185–188.

[47] Y. Yuan, Numerical Methods for Nonlinear Programming, Shanghai Scientific and Technical Publishers, (1993).

[48] Y. Yuan and W. Y. Sun, Optimization Theory and Methods, Science Press, Bijing, (1997).

[49] J. Z. Zhang; N. Y. Deng and L. H. Chen, New quasi-Newton equation and related methods for unconstrained optimization, Journal of Optimization Theory and Applications, vol. 102 (1999) 147–167.

[50] L. Zhang; W. Zhou, and D. H. Li, "A descent modified Polak- Ribiere - Polyak conjugate gradient method and its global convergence," IMA Journal of Numerical Analysis", vol. 26(4) (2006) 629–640.

[51] L. Zhang; W. Zhou, and D. H. Li, "Global convergence of a modified Fletcher-Reeves conjugate gradient method with Armijo-type line search," Numerische Mathematic, vol. 104(4) (2006) 561–572.