COMPUTATIONAL TREATMENTS OF AN IMPROVED CONJUGATE GRADIENT METHOD FOR UNCONSTRAINED MINIMIZATION

*Abbas Y. Al-Bayati & **Walied H. Sharif

*College of Basic Education Telafer, University of Mosul, Mosul, Iraq
**Department of Mathematics and Computer, Foundation Program of Qatar, University of Qatar, Doha, Qatar

(Received on: 05-05-12; Accepted on: 27-05-12)

ABSTRACT

Recently, the authors introduced a new three-terms nonlinear Conjugate Gradient (CG) method [2] for solving unconstrained optimization problems. Their method was compared with the well-known Zhang's three-terms CG-method [32]. This paper contains a description of several new restarting procedures for the same proposed CG-method introduced by the authors and a numerical investigation of the influence of several scaling techniques with a modified perfect cubic line search procedure on their efficiency. Computational results obtained by means of (35) sufficiently difficult problems are given with promising numerical results.

Key Words: Conjugate Gradient Method, Unconstrained Optimization, Convergence Property, Line Searches, Restarting and Scaling Techniques, Large-Scale Problems, Computational Numerical Experiments.


1. INTRODUCTION:

In this study, we are concerned with the minimization of an unconstrained optimization problem to find a local minimum \( x^* \in \mathbb{R}^n \) of the function \( f : X \rightarrow \mathbb{R} \) on an open set \( X \subset \mathbb{R}^n \); i.e. a point \( x^* \in \mathbb{R}^n \) that satisfies the inequality \( f(x^*) \leq f(x) \quad \forall x \in B(x^*, \varepsilon) \) for some \( \varepsilon > 0 \), where \( B(x^*, \varepsilon) = \{ x \in \mathbb{R}^n : \| x - x^* \| < \varepsilon \} \subset X \) is an open ball contained in \( X \subset \mathbb{R}^n \); in other words we want to:

\[
\min \{ f(x) \mid x \in \mathbb{R}^n \}
\]

where \( f : \mathbb{R}^n \rightarrow \mathbb{R} \) is a continuously differentiable function, and its gradient at point \( x_k \) is denoted by \( g_k \) 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:

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

where \( \alpha_k \) is a step-length, and \( d_k \) is a search direction which is determined by:

\[
d_k = \begin{cases} 
-g_0, & \text{if } k = 0, \\
g_k + \beta_k d_{k-1}, & \text{if } k \geq 1,
\end{cases}
\]

where \( \beta_k \) is a scalar and \( d_k \) is a direction vector satisfying the equation \( B d_k + g_k = 0 \), where \( B \) is a symmetric positive definite approximation of the Hessian matrix that is constructed iteratively [18]. If the number of variables is large, then matrix \( B \) cannot be stored, nor factored in a reasonable time, so other methods have to be used. There exist several classes of such methods: Conjugate Gradient (CG) methods [9], difference versions of Truncated Newton (TN) methods [8], Variable Metric (VM) methods with limited storage [20], sparse variants of VM-methods [27], and partitioned VM-methods for separable problems [12]. The last two classes require the special structure of optimization...
problems. From the other classes the simplest are the CG-methods which need only 3-5 n-dimensional vectors (it depends on their implementation). Recently new attention has been given to these methods because they are globally convergent with mild and reasonable assumptions. Their idea starts since 1952, there have been many well-known formulas for the scalar $\beta_k$, for example, Fletcher-Reeves (FR), Polak-Ribiere (PR), Hestenes-Stiefel (HS) and Dai-Yuan (DY) [6, 21]:

\[
\begin{align*}
\beta_k^{FR} &= \frac{g_k^T g_{k-1}}{g_k^T g_{k-1}} \quad , \quad \beta_k^{PR} = \frac{\beta_k^{HS} y_{k-1}}{g_k^T g_{k-1}} \quad , \quad \beta_k^{HS} = \frac{y_k^T y_{k-1}}{d_k^T y_{k-1}} \quad , \quad \beta_k^{DY} = \frac{\|y_k\|^2}{d_k^T y_{k-1}} ,
\end{align*}
\]  
(4)

where $y_{k-1} = g_k - g_{k-1}$, symbol $\| \cdot \|$ denotes the Euclidean norm of vectors. 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 CG-methods defined in (4) have been intensively studied by many researchers [1, 5, 11, 13, 14, 16, 26, 29, 33]. Although the HS method is most general and the FR method is the simplest with good global convergence properties (theoretical), the most numerically efficient was proved to be the PR method. 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 [15]. Common criteria for line search accuracy are the Wolfe conditions [30, 31]:

\[
\begin{align*}
(\delta &\leq \alpha \leq \sigma) \quad \text{where} \quad 10 < \sigma = \delta .
\end{align*}
\]  
(5a)

\[
(\delta \leq \alpha \leq \sigma) \quad \text{where} \quad 0 < \delta \leq \alpha \leq 1 .
\]  
(5b)

It has been shown [7] that for the FR scheme, the strong Wolfe conditions may not yield a direction of descent unless $\sigma \leq 1/2$ . However, The CG-methods are more sensitive to their implementation than the VM methods:

1. The initial estimate $\alpha_0$ of $\alpha_k$ in the line search algorithm does not have theoretical justification for CG-methods. Therefore the CG-methods are more sensitive to the initial estimate $\alpha_0$ than the VM methods.

2. CG-methods need more perfect line search than VM methods. We usually use $\sigma = 0.1$ for CG-VM-methods.

3. CG-methods strongly depend on restarts while VM-methods need not be restarted.

2. PRELIMINARIES:

2.1. Assumption: The objective function $f$ is bounded below, and the level set

\[
F = \{ x \in \mathbb{R}^n \mid f(x) \leq f(x_0) \}
\]  
(6)

2.2. Assumption: In some neighborhood $N$ of $F$, $f$ is differentiable and its gradient is Lipschitz continuous, namely, there exists a positive constant $L$ such that:

\[
\|g(x) - g(y)\| \leq L \|x - y\| \quad \forall x, y \in N
\]  
(7a)

The above assumption implies that there exists a positive constant $\overline{F}$ such that

\[
\|g(x)\| \leq \overline{F}, \quad \forall x \in F
\]  
(7b)

2.3. Zhang’s Three-Terms CG-Method [32]:

Zhang, et al. had introduced a three-term CG method as follows:

\[
d_k = \begin{cases} 
- g_0 , & \text{if } k = 0, \\
- g_k + \beta_k^{DY}_k d_{k-1} - \xi_k (y_{k-1} - T_{k-1} d_{k-1}) , & \text{if } k \geq 1,
\end{cases}
\]  
(8a)

© 2012, IJMA. All Rights Reserved
where 
\[ t \geq 0; \quad \xi_k = \frac{g_k^T d_{k-1}}{d_{k-1}^T y_{k-1}} \quad \text{and} \quad \beta_k^{DL} = \frac{g_k^T (y_{k-1} - ts_{k-1})}{d_{k-1}^T y_{k-1}}. \] (8b)

They show that the sufficient descent condition also holds true if no line search is used, that is,
\[ g_{k-1}^T d_k = -\|g_{k-1}\|^2. \] (9)

In order to achieve the global convergence result, Grippo and Lucidi [13] proposed a new line search. For given constants \( r > 0, \delta > 0, \) and \( \lambda \in (0,1), \) let
\[
\alpha_k = \max \left\{ \lambda^j \left( \frac{r g_k^T d_k}{\|d_k\|} \right), \quad j = 0,1,... \right\} \quad \text{(10a)}
\]
satisfy
\[ f(x_k) \leq f(x_{k-1}) - \delta \alpha_k^2 \|d_{k-1}\|^2. \quad \text{(10b)} \]

This line search will be preferred to the classical Armijo one for the sake of a greater reduction of objective function. Introducing this line search rule, we are now ready to state the outline of the Zhang, et al. [32] first three-term CG-method as follows:

2.4. Outline of Zhang's Three-Terms CG-Algorithm [32]:

---

**Step 1.** Given \( x_0 \in \mathbb{R}^n. \) Let \( 0 < \delta < \sigma < 1, \quad t \geq 0 \) and \( d_0 = -g_0. \) Set \( k := 0. \)

**Step 2.** If \( \|g_k\| \leq 10^{-6}, \) then stop.

**Step 3.** Compute \( d_k \) using (8).

**Step 4.** Find the step-length \( \alpha_k \) satisfying (11) and (12).

\[ f(x_{k-1} + \alpha_k d_k) - f(x_{k-1}) \leq -\delta \alpha_k^2 \|d_{k-1}\|^2, \] (11)

\[ g(x_{k-1} + \alpha_k d_k)^T d_k \geq \sigma g_k^T d_{k-1}. \] (12a)

\[ g_k^T d_{k-1} \leq -\sigma g_{k-1}^T d_{k-1} \] (12b)

and set \( x_k = x_{k-1} + \alpha_k d_k. \)

**Step 5.** Set \( k := k + 1, \) go to Step 2.

2.5. Al-Bayati and Altae Three-Term CG-Method [2]:

The search directions of this method are defined by; see Al-Bayati and Altae:

\[ d_k = \begin{cases} -g_0, & \text{if } k = 0, \\
-g_k + \beta_k^N d_{k-1} - \xi_k \left( y_{k-1} - \frac{\|y_{k-1}\|^2}{s_{k-1}^T y_{k-1}} s_{k-1} \right), & \text{if } k \geq 1,
\end{cases} \] (13a)

where
\[ \beta_k^N = \max \left\{ \frac{g_k^T y_{k-1}}{d_{k-1}^T y_{k-1}}, 0 \right\} - t_k \frac{g_k^T s_{k-1}}{d_{k-1}^T y_{k-1}}, \] (13b)
and
\[ \xi_k = \frac{g^T_k d_{k-1}}{d^T_{k-1} y_{k-1}}; \quad t_k = \frac{\|y_{k-1}\|^2}{s^T_{k-1} y_{k-1}} \geq 0 \] (13c)

It is easy to see that the sufficient descent condition (9) also holds true if no line search is used.

2.6. Outline of Al-Bayati and Altae Three-Term CG-Algorithm [2]:

Step 1. Given \( x_0 \in \mathbb{R}^n \). Let \( 0 < \delta < \sigma < 1, \ t \geq 0 \) and \( d_0 = -g_0 \). Set \( k = 0 \).

Step 2. If \( \|g_k\| \leq 10^{-6} \), then stop.

Step 3. Compute \( d_k \) using (13).

Step 4. Find the step-length \( \alpha_k \) satisfying (11) and (12) and set \( x_k = x_{k-1} + \alpha_k d_k \).

Step 5. Set \( k = k + 1 \), go to Step 2.

3. MODIFIED RESTARTING; SCALING AND LINE SEARCH TECHNIQUES IMPLEMENTED IN THE NEW PROPOSED CG-ALGORITHM:

In this section we are going to introduce several new restarting and several scaling techniques to improve the performance behavior of Al-Bayati and Altae [2] three-terms CG-algorithm and as follows:

3.1. Different Restarting Techniques:

We limit our attention to the PR method, but the same considerations can be used for the HS method. Usually the PR method is implemented with periodic restarts. In [23], Powell points out that the PR method works better if it is restarted whenever
\[ \beta^k_{PR} < 0. \] (14)

The PR method with periodic restarts can be disadvantageous for some problems that require more restarts at the beginning of the iterative process. Also, convergence results noted in this paper that our computational experiments show that the PR method is more efficient if it is restarted not only when (14) holds, but also whenever:
\[ \beta^k_{PR} \leq \eta \beta^k_{FR} \] (15)

and
\[ \lambda \|g_k\|^2 \leq \omega \] (16)

where \( 1 < \eta < 1/(2\sigma) \) is a suitable constant (we recommend \( \lambda = 10^{-8}, \ \eta = 1.34, \ \omega = 10^{-4} \) all recommended values given in this paper were obtained experimentally by means of extensive computations).

3.2. Different Scaling Techniques:

Another useful tool for improving CG methods is scaling, which was originally developed for VM methods [24]. The scaling consists in replacing (3) by:
\[ d_k = \gamma_k \left( -g_k + \beta_k d_{k-1} - t_k s_{k-1} \right), \] (17)

where \( \gamma_k \) is the scaling factor. This type of CG-methods are called spectral CG-methods. Then it found that the best value of this parameter:
\[ \gamma_k = \frac{y^T_{k-1} s_{k-1}}{y^T_{k-1} y_{k-1}}, \] (18)
Note that when we use (17) then PR; FR and DY have to be replaced by:

\[
\beta_{PR} = \frac{1}{y_k} \left( \frac{y_{k-1}^T s_k}{g_{k-1}^T g_k} \right),
\]

(19a)

and

\[
\beta_{FR} = \frac{1}{y_k} \left( \frac{g_k^T g_k}{g_{k-1}^T g_{k-1}} \right),
\]

(19b)

and

\[
\beta_{DY} = \frac{1}{y_k} \left( \frac{d_{k-1}^T y_{k-1}}{x_k^T g_k} \right).
\]

(19c)

For the simplification of subsequent considerations, we have used the following scaling criterion to scale our new proposed three-terms CG-algorithm.

\[
y_k = \bar{\gamma}_1, \quad \text{if} \quad \frac{y_{k-1}^Ts_{k-1}}{y_{k-1}^Ty_{k-1}} < \bar{\gamma}_1
\]

(20a)

\[
y_k = \bar{\gamma}_2, \quad \text{if} \quad \frac{y_{k-1}^Ts_{k-1}}{y_{k-1}^Ty_{k-1}} < \bar{\gamma}_2
\]

(20b)

\[
y_k = \frac{y_{k-1}^Ts_{k-1}}{y_{k-1}^Ty_{k-1}}, \quad \text{otherwise},
\]

(20c)

where \(0 < \bar{\gamma}_1 < 1 < \bar{\gamma}_2\) (we recommend \(\bar{\gamma}_1 = 0.005\) and \(\bar{\gamma}_2 = 200\)). The bounds \(\bar{\gamma}_1\) and \(\bar{\gamma}_2\) serve for improvement of stability.

3.3. Perfect Cubic Line Search Technique:

Since the CG-methods require more perfect line search than other methods, they are very sensitive to its realization. We have essentially used the standard cubic line search implementation; namely (a perfect cubic interpolation), which can be represented by the following algorithm:

3.4. Line Search Algorithm: Input data: \(\Delta > 0; 0 < \phi_1 < \phi_2 < 1\) and \(0 < \delta < \sigma < 1/2\).

\(\delta = 0.0001; \sigma = 0.1, \phi_1 = 0.01, \phi_2 = 0.9\) and \(\Delta = 1000\)

Step (1): Determine the initial estimate \(\alpha_1\) of \(\alpha_k\). This may be taken as: \(\alpha_1 = (1, \text{ or } \min \left(1, 2 \frac{f_{\min} - f_1}{s_1^T g_1} \right))\) and set \((\psi_i = 0, \ i = 1)\)

Step (2): Set \(\alpha_i = \min(\alpha_i, \Delta \| s_k \|)\). Set \(\rho_i = \psi_i\) and \(\psi_i = \alpha_i\). If the conditions (5a) and (5c) are satisfied with \(f_{k+1}\) and \(g_{k+1}\) replaced by \(f(x_k + \alpha_i s_k)\) and \(g(x_k + \alpha_i s_k)\) respectively, then set \(\alpha_k = \alpha_i\) and terminate the computation. If both (5a) and \(s_k^T g(x_k + \alpha_i s_k) < 0\), hold then go to Step(3), else go to Step(4).

Step (3): If \(\alpha_i = \Delta \| s_k \|\) then set \(\alpha_k = \alpha_i\) and terminate the computation, else determine the new estimate \(\alpha_i\) by cubic extrapolation. Set \(\alpha_i = \max(\alpha_i, \psi_i / \phi_2)\), set \(\alpha_i = \max(\alpha_i, \psi_i / \phi_1)\), and go to Step(2).

Step (4): Determine the new estimate \(\alpha_i\) by cubic interpolation. Set \(\alpha_i = \max(\alpha_i, \rho_i + \phi_2 (\psi_i - \rho_i))\), set \(\alpha_i = \max(\alpha_i, \rho_i + \phi_2 (\psi_i - \rho_i))\).
Step (5): If the conditions (5a) and (5c) are satisfied, with $f_{k+1}$ and $g_{k+1}$ replaced by $f(x_k + \alpha_i s_k)$ and $g(x_k + \alpha_i s_k)$ respectively, then set $\alpha_k = \alpha_i$ and terminate the computation. If both (5a) and (5c) hold then set $\rho_i = \alpha_i$, else set $\psi_i = \alpha_i$; go to Step(4).

3.5. Outline of the New Scaled Three-Term CG-Algorithm:

Step 1. Given $x_0 \in \mathbb{R}^n$. Let $0 < \delta < \sigma < 1$, $t \geq 0$ and $d_0 = -g_0$. Set $k := 0$.

Step 2. If $\|g_k\| \leq 10^{-6}$, then stop.

Step 3. Compute $d_k$ using:

$$d_k = \gamma_k \begin{cases} -g_0, & \text{if } k = 0, \\ -g_k + \beta_k d_{k-1} - \left(\frac{g_k^T d_{k-1}}{d_{k-1}^T g_{k-1}}\right) \left(y_{k-1} - \frac{\|y_{k-1}\|^2}{s_{k-1}^T y_{k-1}} s_{k-1}\right), & \text{if } k \geq 1, \end{cases}$$  \hspace{1cm} (21)

$\gamma_k$ is defined by (20) and $\beta_k$ is defined in (13b).

Step 4. Find the step-length $\alpha_k$ satisfying cubic line search Algorithm (3.4) and conditions (11) - (12) and set $x_k = x_{k-1} + \alpha_k d_k$.

Step 5. Do a restart step. If ($\beta_k^{PR} < 0$ or $\beta_k^{PR} \leq \eta \beta_k^{FR}$ or $\lambda \|g_k\|^2 \leq \omega$ ) then set $k = k + 1$, and go to Step 2.

4. CONVERGENCE RESULTS:

It is well known that any CG-method with perfect line search (with (5c) where $\sigma = 0$ ) finds the minimum of a quadratic function after at most n steps. This property implies that any convergent CG-method with asymptotically perfect line search and with periodic restart is n-step quadratically convergent [3]. This result is very useful because asymptotically perfect line search can be easily realized by both quadratic and cubic interpolations. The global convergence of CG methods can be assured by suitable restart rules. The simplest such rule is the so-called angle test which consists in setting $\beta_k = 0$ in (3) whenever:

$$\cos\left(\frac{d_k^T g_k}{\|d_k\| \|g_k\|}\right) < \delta_0$$  \hspace{1cm} (22)

where $\delta_0$ is a prescribed constant (usually $\delta_0 = 10^{-3}$). A more complicated angle test is proposed in [25]. If the line search is asymptotically perfect, the global convergence of CG-methods can be assured by periodic restarts. The first global convergence result which does not depend on restarts has been obtained by Zoutendijk [34] and Powell [22], who proved that the FR method with perfect line search is globally convergent in the sense that

$$\liminf \|g_k\| = 0$$  \hspace{1cm} (23)

where $\liminf \|g_k\|$ is taken over the iterative process (1). Later, Al-Baali [1] generalized this result to include the FR method without perfect line search. He has shown that (22) holds for the FR method whenever $\sigma < 1/2$ in (5). Recently great effort was devoted to generalizing this result to other CG methods. Touati-Ahmed and Storey [28] have shown that the iterative process (2) and (3) with a line search satisfying (5) is globally convergent if:

$$0 \leq \beta_k^{PR} \leq \eta \beta_k^{FR}$$  \hspace{1cm} (24a)
For $1 < \eta < 1/(2\sigma)$. In this case, we have proposed also the following criterion to ensure the global convergence property, namely:

$$\lambda \|g_k\|^2 \leq \omega$$

hold in every iteration, where $0 < \lambda$ and $\omega > 0$ are suitable constants.

The proof given in [28] guarantee that, for $\sigma \eta < 1$, the following inequality:

$$g^T_{k-1}d_{k-1} \leq -\frac{1}{1 - \sigma \eta} \|g_{k-1}\|^2$$

is satisfied at every iteration. Therefore the CG-method is a descent one if (24) holds. The most general result has been obtained by Gilbert and Nocedal [10], who have shown that the both PR and HS methods are globally convergent if they generate positive values of $\beta_k$ and if (25) holds. This result is very important because it allows us to develop a great number of useful restart procedures for CG-methods. The reader may see [19] for more details of some of above theoretical results.

4.1. Lemma: Consider the CG-method in the form (2) and (3), and let the step-length $\alpha_k$ be obtained by the line search Algorithm (3.4) with conditions (11) and (12). Suppose that Assumptions 2.1-2.2 hold. Then one has:

$$\sum_{k=0}^{\infty} \alpha_k^2 \|d_k\|^2 < 0.$$  \hspace{1cm} (26)

Proof: Since $\alpha_k$ is obtained by the line search Algorithm (3.4) with conditions (11)-(12). Then, from (8) and (11) we have

$$f_k - f_{k-1} \leq -\delta \alpha_k^2 \|d_k\|^2 \leq 0.$$  \hspace{1cm} (27)

Hence, $\{f_k\}$ is a decreasing sequence and the sequence $\{\nu_k\}$ is contained in F. Hence, Assumptions 2.1-2.2 imply that there exists a constant $f^*$ such that:

$$\lim_{k \to \infty} f_k = f^*.$$  \hspace{1cm} (28)

From (28), we have:

$$\sum_{k=0}^{\infty} (f_{k-1} - f_k) < +\infty.$$  \hspace{1cm} (29)

This together with (27) implies that (26) holds.

4.2. Lemma: For the new proposed algorithm, defined in (21), if there exists a constant $\varepsilon > 0$ such that:

$$\|g_k\| \geq \varepsilon, \quad \forall k \geq 0,$$  \hspace{1cm} (30)

then there exists a constant $M > 0$ such that

$$\|d_k\| \leq M, \quad \forall k \geq 0.$$  \hspace{1cm} (31)

Proof: The proof is same as in [32] except that we have to prove:

$$I = 2 \frac{\|y_{k-1}\|^2}{s_{k-1}^T y_{k-1}} = 2 \frac{\|y_{k-1}\|^2}{\alpha_{k-1} d_{k-1}^T y_{k-1}} \geq 0$$  \hspace{1cm} (32a)
and

\[ y^T_k \frac{y_k}{y^T_k d_k} \geq 0, \quad \text{since} \quad y^T_k d_k > 0, \quad \text{(B is positive definite)} \]  

(32b)

From the line search conditions (11)-(12) and (8), we have

\[ y^T_k d_k = g^T_k d_k - g^T_k d_k = (1 - \sigma) \| g^T_k d_k \| \leq (1 - \sigma) \| g^T_k d_k \|. \]  

(33)

since \( 0 < \sigma < 1 \) implies \( (1 - \sigma) \) is positive. Since \( \alpha_{k-1} \) is also positive step-size obtained by a line search producer, hence the parameter \( t \) defined in (13c) is positive. This will complete the proof of lemma 4.2. Using the preceding lemmas, we are now ready to give the final convergence results.

4.3. Theorem: Suppose that Assumptions 2.1-2.2 hold. Let \( \{x_k\} \) be a sequence of points generated by the new proposed algorithm defined by (21). Then one has

\[ \liminf_{k \to \infty} \| g_k \| = 0. \]  

(34)

Proof: We proceed by contradiction. Assume that the conclusion is not true, then there exists a positive constant \( \varepsilon \) such that

\[ \| g_k \| \geq \varepsilon, \quad \forall k \geq 0. \]  

(35)

If \( \liminf_{k \to \infty} \alpha_k > 0 \), we have from (26) that \( \liminf_{k \to \infty} \| g_k \| = 0 \). This contradicts assumption (35). Suppose that \( \liminf_{k \to \infty} \alpha_k > 0 \). Using Assumptions 2.1-2.2 and from condition (12a), we obtain:

\[ -(1 - \sigma) g^T_k d_k \leq (g^T_{k+1} - g^T_k) d_k \leq L \alpha_k \| d_k \|^2. \]  

(36)

Combining with (8) yields:

\[ (1 - \sigma) \| g_k \|^2 \leq L \alpha_k \| d_k \|^2. \]  

(37)

The above inequality and Lemma 4.2 imply \( \liminf_{k \to \infty} \| g_k \| = 0 \), which contradicts (35). This completes the proof.

5. NUMERICAL RESULTS.

The main work of this section is to report the performance of the new proposed algorithm (say NEW) on a set of test problems. The codes were written in Fortran77 and in double precision arithmetic. All the tests were performed on a PC. Our experiments were performed on a set of (35) nonlinear unconstrained problems that have second derivatives available. These test problems are contributed in CUTE [4] and their details are given in the Appendix. For each test function we have considered 10 numerical experiments with number of variables \( n = 100, 200, \ldots, 1000 \) and we have reported the total amount of each test problem. In order to assess the reliability of our new proposed method, we have tested it against the proposed Al-Bayati and Altae three-term CG-method [2] using the same test problems. All these methods terminate when the following stopping criterion is met:

\[ \| g_k \| \leq 10^{-6}. \]  

(38)

We also force these routines stopped if the iterations exceed 1000 or the number of function evaluations reach 2000 without achieving convergence. We use Algorithm (3.4) as the line search routine satisfying (11) and (12). Tables 5.1 compares some numerical results for NEW method against Al-Bayati and Al-tae [2] three-term CG-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. In Table 5.2 we have compared the percentage performance of the NEW method against Al-Bayati and Altae three-term CG-method [2] taking over all the tools as 100%.

© 2012, IJMA. All Rights Reserved
Table 5.1. Numerical Results for NEW-Algorithm against Algorithm (2.5):
For the total of (35) test problems

<table>
<thead>
<tr>
<th>n</th>
<th>NOI</th>
<th>NOFG</th>
<th>TIME (seconds)</th>
<th>Gmin</th>
<th>n</th>
<th>NOI</th>
<th>NOFG</th>
<th>TIME (seconds)</th>
<th>Gmin</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>TOTAL 438</td>
<td>805</td>
<td>0.43</td>
<td></td>
<td>1</td>
<td>TOTAL 430</td>
<td>767</td>
<td>0.40</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>TOTAL 358</td>
<td>4643</td>
<td>0.30</td>
<td></td>
<td>2</td>
<td>TOTAL 330</td>
<td>4259</td>
<td>0.22</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>TOTAL 40</td>
<td>90</td>
<td>0.03</td>
<td></td>
<td>3</td>
<td>TOTAL 40</td>
<td>90</td>
<td>0.03</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>TOTAL 4120</td>
<td>6102</td>
<td>2.33</td>
<td></td>
<td>4</td>
<td>TOTAL 2809</td>
<td>4203</td>
<td>2.49</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>TOTAL 511</td>
<td>5846</td>
<td>0.42</td>
<td></td>
<td>5</td>
<td>TOTAL 504</td>
<td>5801</td>
<td>0.40</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>TOTAL 215</td>
<td>436</td>
<td>0.07</td>
<td></td>
<td>6</td>
<td>TOTAL 232</td>
<td>444</td>
<td>0.06</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>TOTAL 335</td>
<td>6178</td>
<td>2.96</td>
<td></td>
<td>7</td>
<td>TOTAL 301</td>
<td>3590</td>
<td>1.44</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>TOTAL 143</td>
<td>293</td>
<td>0.04</td>
<td></td>
<td>8</td>
<td>TOTAL 198</td>
<td>361</td>
<td>0.05</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>TOTAL 40</td>
<td>90</td>
<td>0.06</td>
<td></td>
<td>9</td>
<td>TOTAL 40</td>
<td>90</td>
<td>0.06</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>TOTAL 245</td>
<td>444</td>
<td>0.03</td>
<td></td>
<td>10</td>
<td>TOTAL 247</td>
<td>440</td>
<td>0.04</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>TOTAL 211</td>
<td>2104</td>
<td>0.91</td>
<td></td>
<td>11</td>
<td>TOTAL 213</td>
<td>2170</td>
<td>0.98</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>TOTAL 615</td>
<td>967</td>
<td>0.20</td>
<td></td>
<td>12</td>
<td>TOTAL 537</td>
<td>898</td>
<td>0.18</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>TOTAL 20</td>
<td>50</td>
<td>0.02</td>
<td></td>
<td>13</td>
<td>TOTAL 20</td>
<td>50</td>
<td>0.02</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>TOTAL 109</td>
<td>223</td>
<td>0.04</td>
<td></td>
<td>14</td>
<td>TOTAL 102</td>
<td>208</td>
<td>0.03</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>TOTAL 111</td>
<td>232</td>
<td>0.06</td>
<td></td>
<td>15</td>
<td>TOTAL 111</td>
<td>232</td>
<td>0.06</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>TOTAL 149</td>
<td>272</td>
<td>0.07</td>
<td></td>
<td>16</td>
<td>TOTAL 148</td>
<td>270</td>
<td>0.07</td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>TOTAL 473</td>
<td>910</td>
<td>0.10</td>
<td></td>
<td>17</td>
<td>TOTAL 503</td>
<td>983</td>
<td>0.10</td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>TOTAL 661</td>
<td>8083</td>
<td>0.75</td>
<td></td>
<td>18</td>
<td>TOTAL 580</td>
<td>6812</td>
<td>0.58</td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>TOTAL 149</td>
<td>394</td>
<td>0.03</td>
<td></td>
<td>19</td>
<td>TOTAL 149</td>
<td>394</td>
<td>0.03</td>
<td></td>
</tr>
</tbody>
</table>
Table 5.2. Percentage performance of NEW-Algorithm against Algorithm (2.5)

<table>
<thead>
<tr>
<th>Tools</th>
<th>Algorithm (2.5)</th>
<th>NEW (Algorithm 3.5)</th>
</tr>
</thead>
<tbody>
<tr>
<td>NOI</td>
<td>100%</td>
<td>78.5%</td>
</tr>
<tr>
<td>NOFG</td>
<td>100%</td>
<td>84.5%</td>
</tr>
<tr>
<td>TIME</td>
<td>100%</td>
<td>80.3%</td>
</tr>
</tbody>
</table>
It is clear from Table (5.2) that taking, over all, the Tools as a 100% for the Al-Bayati and Altae three-term CG-method, namely algorithm (2.5), the NEW-Algorithm has an improvement, in about (21.5%) NOI; (16.5%) NOFG and (19.7%) TIME.

CONCLUSIONS:

Taking everything into consideration the new proposed scaled three-term CG-method have been obtained very significant development as we have expected, we think that, for all the specific problems, the enhancement of the new proposed method is very robust. However, we know that CG-methods are sensitive to the order of interpolation; therefore, we have recommend a modified perfect cubic interpolation over the standard quadratic one in our implementations. Hence, we believe that the new method is a valid approach for the problems and has its own potential. Also, the effectiveness of this new proposed method depends on the robustness set of selected scaling criteria and several selected sets of restating techniques used in this research besides the selected set of test functions.

ACKNOWLEDGEMENTS

We are indebted to B. Albaro for kindly sharing with us her findings in this subject and completing the layout of this manuscript.

APPENDIX

The details of the test functions, used in this paper, can be found in [4]. The numbers (1-35) in Table 5.1 indicate to:

1- Extended Trigonometric Function.
2- Extended Penalty Function.
3- Raydan 2 Function.
4- Diagonal2 Function.
5- Generalized Tri-diagonal-1 Function.
6- Extended Tri-diagonal-1 Function.
7- Extended 3-Exponential Terms Function.
8- Diagonal4 Function.
9- Diagonal5 Function.
10- Extended Himmelblau Function.
11- Extended PSC1 Function.
12- Extended Block Diagonal BD1 Function.
13- Extended EP1 Function.
14- DIXMAANA CUTE- Function.
15- DIXMAANB CUTE- Function.
16- DIXMAANC CUTE- Function.
17- Broyden Tri-diagonal Function.
18- EDENSCH CUTE- Function.
19- VARDIM CUTE- Function.
20- LIARWHD CUTE- Function.
21- DIAGONAL 6 Function.
22- ENGVAl1 CUTE- Function.
23- DENSCHNA CUTE- Function.
24- DENSCHNB CUTE- Function.
25- DENSCHNF CUTE- Function.
26- Generalized Quartic GQ1 function.
27- Diagonal 7 Function.
28- Diagonal 8 Function.
29- Full Hessian Function.
30- SINCOS Function.
31- Generalized quartic GQ2 function.
32- ARGLINB CUTE-Function.
33- FLETCHCR CUTE-Function.
34- HIMMELBG CUTE-Function.
35- HIMMELBH CUTE-Function.

REFERENCES

18. L. Luk’an, Computational experience with improved variable metric methods for unconstrained minimization, Kybernetika, 26 (1990), pp. 415-431.

Source of support: Nil, Conflict of interest: None Declared