Different Types of Three-Term CG-Methods with Sufficient Descent and Conjugacy Conditions

Abbas Y. Al-Bayati* and Hawraz N. Al-Khayat**

* College of Telafer Basic Education, Mathematics, Mosul University, Iraq,
E-mail: profabbasalbayati@yahoo.com
** College of Computer Sciences and Mathematics, Mathematics, Mosul University, Iraq,
E-mail:hawraz_na@yahoo.com.

Received: 6 Mar. 2014, Revised: 3 Apr. 2014, Accepted: 24 April 2014
Published online: 7 June 2014

Abstract:

It is very important to generate a descent search direction independent of line searches in showing the global convergence of conjugate gradient methods. Recently, Zhang et al. proposed a three-term of PR method (TTPR) and HS method (TTHS), both of which can produce sufficient descent condition. In this paper, we treat two subjects: we first consider new unified formula of three-term CG algorithm, second we suggested new scaled three-term algorithm based on Birgin-Martínez algorithm and which satisfied both the descent and conjugacy conditions are proposed. This algorithms are modification of the Hestenes-Stiefel and Birgin-Martínez algorithms, also the algorithms could be considered as a modification of the memoryless BFGS quasi-Newton method. Our algorithms can proved the global convergence property and more efficiently than HS and BM algorithms in numerical results.

Keywords: Three-Term Conjugate Gradient, Global Convergence, Unconstrained Optimization, Descent Direction, Conjugacy Condition, Memoryless BFGS.

1. Introduction

In this paper, we consider the unconstrained optimization problem

$$\min_{x \in R^n} f(x), \qquad (1.1)$$

Where $f : R^n \to R$ is continuously differentiable. Conjugate gradient methods (CG) are quite useful for solving the problem (1.1), especially for large-scale problems because they do not require the storage of matrices. The CG methods have the following form:

$$x_{k+1} = x_k + \alpha_k d_k \qquad (1.2)$$

$$d_{k+1} = -g_{k+1} + \beta_k d_k \;, k \ge 0 \;, \; d_0 = -g_0 \qquad (1.3)$$

We denote $g_k = \nabla f(x_k)$ as the gradient of $f(x)$ at the current iteration x_k, d_k is the search direction, α_k is positive the step-length obtain by line search procedure, x_{k+1} is the next iteration and β_k a scalar given by different formula which result in distinct CG methods. The CG methods differ in the way of selecting β_k. Several well-known formulae are given by:

$$\beta_k^{HS} = \frac{g_{k+1}^T y_k}{d_k^T y_k}, \qquad (1.4)$$

(Hestenes and Stiefel [15])

$$\beta_k^{FR} = \frac{\|g_{k+1}\|^2}{\|g_k\|^2},$$

(Fletcher and Reeves [18]) (1.5)

$$\beta_k^{PR} = \frac{g_{k+1}^T y_k}{\|g_k\|^2},$$

(Polak and Ribière [4]) (1.6)

$$\beta_k^{DY} = \frac{\|g_{k+1}\|^2}{d_k^T y_k},$$

(Dai and Yuan [21]) (1.7)

$$\beta_k^{BM} = \frac{g_{k+1}^T(\theta_k y_k - s_k)}{y_k^T s_k},$$

(Birgin and Martínez [3]) (1.8)

where $\|\ .\ \|$ denotes the Euclidean norm, $y_k = g_{k+1} - g_k$ and $s_k = x_{k+1} - x_k$. There are numerous research on convergence properties of these methods, e.g. [8–20]. Note that these formulas for β_k are equivalent to each other if the objective function is a strictly convex quadratic function and α_k is the one-dimensional minimizer. For non-quadratic functions, each choice for the parameter β_k leads to very different performance of the corresponding algorithms.

One of the first general three-term conjugate gradient methods was proposed by Beale [6] as

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

(1.9)

Where $\beta_k = \beta_k^{HS} (or\ \beta_k^{FR}, \beta_k^{PR}\ etc.)$

$$\gamma_k = \begin{cases} 0, & k = t+1 \\ \dfrac{g_{k+1}^T y_t}{d_t^T y_t}, & k > t+1 \end{cases}$$

and d_t is a restart direction. Nazareth [11] proposed a conjugate gradient algorithm using a three-term recurrence formula:

$$d_{k+1} = -y_k + \frac{y_k^T y_k}{y_k^T d_k} d_k + \frac{y_{k-1}^T y_k}{y_{k-1}^T d_{k-1}} d_{k-1}$$

(1.10)

With $d_{-1}=0$, $d_0=0$. Recently, Zhang et al. [12,13] proposed a three-term PR conjugate gradient method (TTPR) and a descent modified HS conjugate gradient method with three-terms (TTHS) respectively, that is,

$$d_{k+1}^{TTPR} = -g_{k+1} + \beta_k^{PR} d_k - \theta_k^{(1)} y_k$$

(1.11)

$$d_{k+1}^{TTHS} = -g_{k+1} + \beta_k^{HS} s_k - \theta_k^{(2)} y_k$$

(1.12)

Where

$$\theta_k^{(1)} = \frac{g_{k+1}^T d_k}{\|g_k\|^2}$$

(1.13)

And

$$\theta_k^{(2)} = \frac{g_{k+1}^T s_k}{s_k^T y_k}$$

(1.14)

Zhang et al. [8] introduced another three-term conjugate gradient method based of the Dai–Liao method[22] as follows:

$$d_{k+1} = -g_{k+1} + \frac{g_{k+1}^T(y_k - ts_k)}{d_k^T y_k} d_k - \frac{g_{k+1}^T d_k}{d_k^T y_k}(y_k - ts_k)$$

(1.15)

Where d_0=-g_0 and $t \geq 0$, it is easy to see that the sufficient descent condition also holds independent by the line search, i.e. for this method $g_k^T d_k = -\|g_k\|^2$ for all k. A specialization of this three-term CG method given by (1.15) was developed by Al-Bayati and Sharif [1], where the search direction is computed as

$$d_{k+1} = -g_{k+1} + \beta_k^{DL+} s_k - \frac{g_{k+1}^T s_k}{\|g_k\|^2}(y_k - t s_k)$$

(1.16)

Where $\beta_k^{DL+} = \max\left\{\frac{y_k^T g_{k+1}}{y_k^T s_k}, 0\right\} - t\frac{s_k^T g_{k+1}}{y_k^T s_k}$

and $t = 2\frac{\|y_k\|^2}{y_k^T s_k}$. It is easy to see that (1.16) satisfies the sufficient descent condition independent of the line search used see [1].

The line search in the conjugate gradient algorithms is often based on the standard Wolfe conditions:

$$f(x_k + \alpha_k d_k) - f(x_k) \le \delta \alpha_k g_k^T d_k, \quad (1.17)$$

$$g_{k+1}^T d_k \ge \sigma g_k^T d_k \quad (1.18)$$

where $0 < \delta \le \sigma < 1$, However, for some conjugate gradient algorithms, a stronger version of the Wolfe line search conditions, given by (1.17) and

$$\left|g_{k+1}^T d_k\right| \le -\sigma g_k^T d_k \quad (1.19)$$

In this paper, we mainly treat two subjects: First, we give the global convergence theorem of the three-term conjugate gradient method with a unified formula of the new direction but 3 different type of the positive parameters t_k which is obtained by a modification of the BFGS updating scheme of the inverse approximation of the function f restarted as the identity matrix at every step. This computational scheme is a modification of the three-term of Zhang et al[13]. in (1.12). Second, in order to accelerate the three-term CG algorithm, we apply the formula of β_k^{BM} to the scaled three-term CG method. We show the global convergence property of the proposed methods.

In this paper is organized as follows: in **Section 2** we give the new two formulas of three-term CG algorithms. In **Section 3** we proved this new search directions are satisfied both sufficient descent and conjugacy conditions. Moreover, our algorithms are proved the global convergence property satisfied with Wolfe line search conditions in **Section 4**. In **Section 5**, some numerical results and comparisons with some other three-term CG algorithms are presented. It is shown that the performances of this new CG algorithms being faster.

2. New Three Term Conjugate Gradient methods:

We divided this section into two part the first for described our simple three-term CG algorithms (**TTHS**) in **2.1**, **2.2** and **2.3** for which , independent of the line search, at every step both the descent condition and the conjugacy condition are satisfied. We say the three first algorithms can sum in the unified formula such that:

$$d_{k+1} = -g_{k+1} + \beta_k^{HS} d_k + \varphi_k (y_k - t_i s_k), \; i = 1,2,3$$

In the second part we suggested new type of scaled three-term CG algorithm in the final of this section i.e.

$$d_k = -\theta_k g_k + \beta_k^{BM} s_{k-1} - \eta_k p_k$$

the same way this method is satisfied descent and conjugacy conditions.

2.1. First HS Three Term Conjugate Gradient Methods(1TTHS):

The algorithm is given by (1.2), where the direction d_{k+1}^{new1} is computed as

$$d_{k+1}^{new1} = -g_{k+1} + \beta_k^{HS} d_k + \varphi_k (y_k - t_1 s_k) \quad (2.1)$$

$$\varphi_k = -\frac{s_k^T g_{k+1}}{y_k^T s_k}, \quad (2.2)$$

$$t_1 = \left[1 + \frac{\|y_k\|^2}{y_k^T s_k}\right] \quad (2.3)$$

Observe that the direction d_{k+1} from (2.1) can be written as

$$d_{k+1} = -Q_{k+1}^{(1)} g_{k+1} \quad (2.4)$$

where the matrix $Q_{k+1}^{(1)}$ is given by

$$Q_{k+1}^{(1)} = I - \frac{s_k y_k^T - y_k s_k^T}{y_k^T s_k} + \left[1 + \frac{\|y_k\|^2}{y_k^T s_k}\right] \frac{s_k s_k^T}{y_k^T s_k} \quad (2.5)$$

As we know, the BFGS updating of the inverse approximation of the Hessian of function f is:

$$H_{k+1} = H_k - \left[\frac{s_k y_k^T H_k + H_k y_k s_k^T}{y_k^T s_k}\right] + \left[1 + \frac{y_k^T H_k y_k}{y_k^T s_k}\right] \frac{s_k s_k^T}{y_k^T s_k} \quad (2.6)$$

Obviously, the matrix $Q_{k+1}^{(1)}$ in (2.5) is a modification of the BFGS updating (2.6) in the sense that it is restarted with the identity matrix at every step ($H_k = I$), and more importantly the sign in front of $y_k s_k^T$ in the second term of (2.5) is modified to get the descent property, as is proved in the following proposition.

We can made a modification of (2.1) by

$$d_{k+1} = -g_{k+1} + \beta_k^{HS+} d_k + \varphi_k (y_k - t_1 s_k) \quad (2.7)$$

$$d_{k+1} = -g_{k+1} + \max\{\frac{g_{k+1}^T y_k}{d_k^T y_k}, 0\} d_k + \varphi_k (y_k - t_1 s_k) \quad (2.8)$$

In this paper we use the follows acceleration scheme to improvement this new four proposed algorithms against to the (BM-CG, BW-CG and TTHS-CG methods)

2.1.1. An Acceleration Scheme of the Line Search Parameter.

In [7] Nocedal pointed out that in CG-methods the step lengths may differ from 1 in a very unpredictable manner. They can be larger or smaller than 1 depending on how the problem is scaled. This is in very sharp contrast to the Newton and QN-methods, including the limited memory QN-methods, which accept the unit step-length most of the time along the iterations, and therefore usually they require only few function evaluations per search direction. Numerical comparisons between CG-methods and the limited memory QN-method by Liu and Nocedal [2], show that the latter is more successful [24]. One explanation of efficiency of this limited memory QN-method is given by its ability to accept unity step-lengths along the iterations. In this section we take advantage of this behavior of CG-algorithms and consider an acceleration scheme which was presented in [16]. In accelerated algorithm instead of (1.2) the new estimation of the minimum point is computed as:

$$x_{k+1} = x_k + \lambda_k \alpha_k d_k \quad (2.9a)$$

Where $\lambda_k = -\frac{a_k}{b_k}$ (2.9b)

$a_k = \alpha_k g_k^T d_k$, $b_k = -\alpha_k (g_k - g_z)^T d_k$, $g_z = \nabla f(z)$ and $z = x_k + \alpha_k d_k$. Hence, if $b_k \neq 0$, then the new estimation of the solution is computed as $x_{k+1} = x_k + \lambda_k \alpha_k d_k$, otherwise

$x_{k+1} = x_k + \alpha_k d_k$. Therefore, using the definitions of g_k, s_k, y_k and the above acceleration scheme (2.9) we can present the following hybrid CG-algorithm.

Algorithm 1TTHS

Step1. Given $x_0 \in R^n$, Let $0 < \delta < \sigma < 1$, $t \ge 0$ and $d_0 = -g_0$. Set $k = 0$.

Step2. If stopping criteria ($\|g_k\|_\infty \le 10^{-6}$) satisfied, then stop.

Step3. Find the step-length α_k s.t. $\alpha_k > 0$ satisfying the Wolfe line search condition and compute $z = x_k + \alpha_k d_k$,

$y_k = g_k - g_z$, $g_z = \nabla f(z)$.

Acceleration scheme: compute, $a_k = \alpha_k g_k^T d_k$, $b_k = -\alpha_k y_k^T d_k$, If $b_k \ne 0$, then compute, $\lambda_k = -\frac{a_k}{b_k}$ and update the variables as $x_{k+1} = x_k + \lambda_k \alpha_k d_k$, otherwise update the variables as $x_{k+1} = x_k + \alpha_k d_k$.

Step4. Determine the parameter φ_k and t_1 as in (2.2) and (2.3), respectively.

Step5. Compute the new 1 search direction d_{k+1}^{new1} using (2.1)

Step6. If Powell restart criterion i.e. $|g_{k+1}^T g_k| > 0.2\|g_{k+1}\|^2$ then set $d_{k+1}^{new1} = -g_{k+1}$

Step7. Set $k = k+1$, go to Step 2.

2.2. Second HS Three Term Conjugate Gradient Methods(2TTHS):

The algorithm is given by (1.2), where the new search direction d_{k+1}^{new2} is computed as

$$d_{k+1}^{new2} = -g_{k+1} + \beta_k^{HS} d_k + \varphi_k (y_k - t_2 s_k) \quad (2.10)$$

$$\varphi_k = -\frac{s_k^T g_{k+1}}{y_k^T s_k}, \quad (2.11a)$$

$$t_2 = \left[1 + 2\frac{\|y_k\|^2}{y_k^T s_k}\right] \quad (2.11b)$$

Observe that the direction d_{k+1} from (2.10) can be written as

$$d_{k+1} = -Q_{k+1}^{(1)} g_{k+1} \quad (2.12)$$

where the matrix $Q_{k+1}^{(1)}$ is given by (2.5),As we know, the BFGS updating of the inverse approximation of the Hessian of function f is

$$H_{k+1} = H_k - \left[\frac{s_k y_k^T H_k + H_k y_k s_k^T}{y_k^T s_k}\right] + \left[1 + \frac{y_k^T H_k y_k}{y_k^T s_k}\right]\frac{s_k s_k^T}{y_k^T s_k} \quad (2.13)$$

We can made a modification of (2.10) by

$$d_{k+1} = -g_{k+1} + \beta_k^{HS+} d_k + \varphi_k (y_k - t_2 s_k)$$

$$d_{k+1} = -g_{k+1} + \max\{\frac{g_{k+1}^T y_k}{d_k^T y_k}, 0\} d_k + \varphi_k (y_k - t_2 s_k)$$

Algorithm 2TTHS

Step1. Given $x_0 \in R^n$, Let $0 < \delta < \sigma < 1$, $t \geq 0$ and $d_0 = -g_0$. Set $k = 0$.

Step2. If stopping criteria ($\|g_k\|_\infty \leq 10^{-6}$) satisfied, then stop.

Step3. Find the step-length α_k s.t. $\alpha_k > 0$ satisfying the Wolfe line search condition and compute $z = x_k + \alpha_k d_k$, $y_k = g_k - g_z$, $g_z = \nabla f(z)$.
Acceleration scheme: compute, $a_k = \alpha_k g_k^T d_k$, $b_k = -\alpha_k y_k^T d_k$, If $b_k \neq 0$, then compute, $\lambda_k = -\frac{a_k}{b_k}$ and update the variables as $x_{k+1} = x_k + \lambda_k \alpha_k d_k$, otherwise update the variables as $x_{k+1} = x_k + \alpha_k d_k$.

Step4. Determine the parameter φ_k and t_2 as in (2.11).

Step5. Compute the new 2 search direction d_{k+1}^{new2} using (2.10)

Step6. If Powell restart criterion i.e. $|g_{k+1}^T g_k| > 0.2\|g_{k+1}\|^2$ then set $d_{k+1}^{new2} = -g_{k+1}$

Step7. Set $k = k+1$, go to Step 2.

2.3. Third HS Three Term Conjugate Gradient Methods(3TTHS):

$$d_{k+1}^{new3} = -g_{k+1} + \beta_k^{HS} s_k + \varphi_k (y_k - t_3 s_k)$$
(2.14) Where

$$\varphi_k = \frac{s_k^T g_{k+1}}{s_k^T y_k},$$
(2.15)

And

$$t_3 = \frac{2\|y_k\|^2}{s_k^T y_k}$$
(2.16)

Al-Bayati [23] investigated another family of QN-method for which the updating matrix H_{k+1} was defined by:

$$H_{k+1} = H_k + \left[\frac{2y_k^T H_k y_k}{(s_k^T y_k)^2}\right] s_k s_k^T - \left[\frac{H_k y_k s_k^T + s_k y_k^T H_k}{y_k^T s_k}\right] \quad (2.17)$$

If $H_k = I$ then the above matrix is defined by:

$$Q_{k+1}^{(2)} = I + \left[\frac{2y_k^T y_k}{(s_k^T y_k)^2}\right] s_k s_k^T - \frac{y_k s_k^T + s_k^T y_k}{s_k^T y_k}$$
(2.18)

Since $d_{k+1} = -Q_{k+1}^{(2)} g_{k+1}$

$$d_{k+1} = -g_{k+1} - \left[\frac{2y_k^T y_k}{s_k^T y_k} \frac{s_k^T g_{k+1}}{s_k^T y_k}\right] s_k + \frac{s_k^T g_{k+1}}{s_k^T y_k} y_k + \frac{y_k^T g_{k+1}}{s_k^T y_k} s_k$$

$$d_{k+1} = -g_{k+1} + \frac{y_k^T g_{k+1}}{s_k^T y_k} s_k + \frac{s_k^T g_{k+1}}{s_k^T y_k} y_k - \frac{s_k^T g_{k+1}}{s_k^T y_k}\left[\frac{2y_k^T y_k}{s_k^T y_k}\right] s_k$$

$$d_{k+1} = -g_{k+1} + \beta_k^{HS} s_k + \frac{s_k^T g_{k+1}}{s_k^T y_k}\left(y_k - \frac{2y_k^T y_k}{s_k^T y_k} s_k\right) \quad (2.19)$$

$$d_{k+1}^{new3} = -g_{k+1} + \beta_k^{HS} s_k + \varphi_k (y_k - t_3 s_k)$$
(2.20)

Where

$$\varphi_k = \frac{s_k^T g_{k+1}}{s_k^T y_k} \quad \text{and} \quad t_3 = \frac{2\|y_k\|^2}{s_k^T y_k}$$

And also we can use the β_k^{HS+} instead of β_k^{HS} in (2.21):

$$d_{k+1} = -g_{k+1} + \beta_k^{HS+} d_k + \varphi_k (y_k - t_3 s_k)$$

$$d_{k+1} = -g_{k+1} + \max\{\frac{g_{k+1}^T y_k}{d_k^T y_k}, 0\} d_k + \varphi_k (y_k - t_3 s_k) \quad (2.21)$$

Algorithm 3TTHS

Step1. Given $x_0 \in R^n$, Let $0 < \delta < \sigma < 1$, $t \geq 0$ and $d_0 = -g_0$. Set $k = 0$.

Step2. If stopping criteria ($\|g_k\|_\infty \leq 10^{-6}$) satisfied, then stop.

Step3. Find the step-length α_k s.t. $\alpha_k > 0$ satisfying the Wolfe line search condition and compute
$z = x_k + \alpha_k d_k$,
$y_k = g_k - g_z$, $g_z = \nabla f(z)$.
Acceleration scheme: compute, $a_k = \alpha_k g_k^T d_k$, $b_k = -\alpha_k y_k^T d_k$, If $b_k \neq 0$, then compute, $\lambda_k = -\frac{a_k}{b_k}$ and update the variables as $x_{k+1} = x_k + \lambda_k \alpha_k d_k$, otherwise update the variables as $x_{k+1} = x_k + \alpha_k d_k$.

Step4. Determine the parameter φ_k and t_3 as in (2.15) and (2.16), respectively.

Step5. Compute the new 3 search direction d_{k+1}^{new3} using (2.14)

Step6. If Powell restart criterion i.e. $|g_{k+1}^T g_k| > 0.2\|g_{k+1}\|^2$ then set $d_{k+1}^{new3} = -g_{k+1}$

Step7. Set $k = k+1$, go to Step 2.

2.4. New Scaled Three Term Conjugate Gradient Method (4TTCG)

In this part, we describe our scaled three-term conjugate gradient algorithm for which, independent of the line search, at every step both the descent condition and the conjugacy condition are satisfied. The algorithm is given by (1.2), where the new search direction d_{k+1}^{new4} is computed as

$$d_{k+1}^{new4} = -\theta_k g_{k+1} + \beta_k^{BM} s_k - \eta_k p_k \quad (2.22)$$

Where $$\beta_k^{BM} = \frac{g_{k+1}^T(\theta_k y_k - s_k)}{y_k^T s_k}, \quad (2.23)$$

$$p_k = \theta_k y_k - s_k \quad , \quad (2.24)$$

$$\theta_k = \frac{s_k^T s_k}{y_k^T s_k}, \quad \eta_k = \frac{s_k^T g_{k+1}}{y_k^T s_k} \quad (2.25)$$

The new search direction can be written as follows:

$$d_{k+1}^{new4} = -\theta_k g_{k+1} + \frac{g_{k+1}^T(\theta_k y_k - s_k)}{y_k^T s_k} s_k - \frac{s_k^T g_{k+1}}{y_k^T s_k}(\theta_k y_k - s_k) \quad (2.26)$$

Algorithm 4TTCG

Step1. Select a starting point $x_0 \in R^n$, Let $0 < \delta < \sigma < 1$, $t \geq 0$ and $d_0 = -g_0$. Set $k = 0$.

Step2. If stopping criteria ($\|g_k\|_\infty \leq 10^{-6}$) satisfied, then stop.

Step3. Find the step-length α_k s.t. $\alpha_k > 0$ satisfying the Wolfe line search condition and compute
$z = x_k + \alpha_k d_k$,
$y_k = g_k - g_z$, $g_z = \nabla f(z)$.
Acceleration scheme: compute, $a_k = \alpha_k g_k^T d_k$, $b_k = -\alpha_k y_k^T d_k$, If

$b_k \neq 0$, then compute, $\lambda_k = -\frac{a_k}{b_k}$ and update the variables as $x_{k+1} = x_k + \lambda_k \alpha_k d_k$, otherwise update the variables as $x_{k+1} = x_k + \alpha_k d_k$.

Step4. Determine the parameter θ_k, η_k and p_k as in (2.25) and (2.24), respectively.

Step5. Compute the conjugate parameter β_k^{BM} in (2.23).

Step6. Compute the new scaled search direction d_{k+1}^{new4} using (2.22).

Step7. If Powell restart criterion i.e. $\left| g_{k+1}^T g_k \right| > 0.2 \|g_{k+1}\|^2$ then set $d_{k+1}^{new4} = -g_{k+1}$.

Step8. Set $k = k+1$, go to Step 2.

3. Descent Direction and Conjugacy Conditions for New TTCG methods

As mentioned in Section 1 many researchers have studied three term CG methods which satisfy the sufficient descent and conjugacy conditions based on HS[15] and BM[3] methods. This section is important to prove the new proposed algorithms are global convergence with using Wolfe line search conditions (1.17) and (1.19) in the next section.

The following two propositions implies that the first new three algorithms are satisfied the sufficient descent and conjugacy property.

Proposition 3.1.

Suppose that the line search satisfies the Strong Wolfe conditions (1.17) and (1.19), then d_{k+1}^{new} where given by (2.1), (2.10) and (2.14) all of this are descent direction.

$$g_{k+1}^T d_{k+1} \leq 0$$

Proof.

Case 1: If the first new search direction d_{k+1}^{new1} is used and since the line search satisfied the Wolfe conditions, it follows that $y_k^T s_k > 0$. Now, by direct computation, we have:

$$g_{k+1}^T d_{k+1} = -\|g_{k+1}\|^2 - \left[1 + \frac{\|y_k\|^2}{y_k^T s_k}\right] \frac{(s_k^T g_{k+1})^2}{y_k^T s_k} \leq 0 \quad (3.1)$$

Case 2: If the d_{k+1}^{new2} and Wolfe line search used in this method, it follows that $y_k^T s_k > 0$. Now, by some algebraic computation, we have:

$$g_{k+1}^T d_{k+1} = -\|g_{k+1}\|^2 - \left[1 + 2\frac{\|y_k\|^2}{y_k^T s_k}\right] \frac{(s_k^T g_{k+1})^2}{y_k^T s_k} \leq 0 \quad (3.2)$$

Case 3: Similar to the previous two cases and under the same conditions, we can say that, it follows that $y_k^T s_k > 0$. Now, by some algebraic computation, we have:

$$g_{k+1}^T d_{k+1} = -\frac{1}{2}\|g_{k+1}\|^2 \leq 0 \quad (3.3)$$

So we concluded that the three cases of the proposed algorithms with three term CG give us the descent direction.

Proposition 3.2.

If that the line search satisfies the Standard Wolfe condition, then d_{k+1}^{new} where given by (2.1), (2.10) and (2.14) all of this are satisfied Dai-Liao conjugacy condition[22]

$$y_k^T d_{k+1} = -v_k (s_k^T g_{k+1}) \quad (3.4)$$

where $v_k > 0$ for all k.

Proof.

Case 1: If the first new search direction d_{k+1}^{new1} is used and by direct computation, we get:

$$y_k^T d_{k+1} = -(1+2\frac{\|y_k\|^2}{y_k^T s_k})(s_k^T g_{k+1}) \equiv -v_k (s_k^T g_{k+1}) \quad (3.5)$$

Where $v_k = (1+2\frac{\|y_k\|^2}{y_k^T s_k}) > 0$, since $y_k^T s_k > 0$.

Case 2: If the d_{k+1}^{new2}, By direct computation, we get:

$$y_k^T d_{k+1} = -(1+3\frac{\|y_k\|^2}{y_k^T s_k})(s_k^T g_{k+1}) = -v_k (s_k^T g_{k+1}) \quad (3.6)$$

Where $v_k = \left[1+3\frac{\|y_k\|^2}{y_k^T s_k}\right] > 0$, since $y_k^T s_k > 0$.

Case 3: Similar to the previous two cases, by direct computation, we get:

$$y_k^T d_{k+1} = -2\frac{\|y_k\|^2}{y_k^T s_k}(s_k^T g_{k+1}) = -v_k (s_k^T g_{k+1}) \quad (3.7)$$

Where $v_k = 2\frac{\|y_k\|^2}{y_k^T s_k} > 0$, since $y_k^T s_k > 0$.

Observe that, if f is strongly convex or the line search satisfies the Wolfe conditions (1.17) and (1.19), then $y_k^T s_k > 0$, and therefore the computational scheme above yields descent. Besides, the direction d_{k+1}^{new} satisfies the Dai–Liao conjugacy condition (3.4), where $v_k > 0$ at every iteration. On the other hand, if the line search is exact, i.e. $s_k^T g_{k+1} = 0$, then (2.1) , (2.10) and (2.14) reduces to the Hestenes–Stiefel method.

Now, we take the new algorithm for the fourth to prove at the same way as the previous two propositions.

Proposition 3.3.
Suppose that the line search satisfies the Strong Wolfe conditions (1.17) and (1.19), then d_{k+1}^{new4} where given by (2.26) is sufficient descent direction.

Proof.
If the new search direction d_{k+1}^{new4} is used and since the line search satisfied the Wolfe conditions, it follows that $y_k^T s_k \neq 0$. Now, By direct computation, we have:

$$g_{k+1}^T d_{k+1} = -\frac{s_k^T s_k}{y_k^T s_k}\|g_{k+1}\|^2 \quad (3.8)$$

Observe that the Wolfe condition gives $s_k^T s_k > 0$ Hence, $s_k^T s_k / y_k^T s_k > 0$ for any k. Therefore, for all $k \geq 0$, the search direction (2.26) satisfies a variant of the sufficient descent condition $g_{k+1}^T d_{k+1} = -c\|g_{k+1}\|^2$, where $c > 0$ is modified at every iteration.

Proposition 3.4.
Direction d_{k+1}^{new4} given by (2.26) satisfies the Dai–Liao[22] conjugacy condition.

Proof.

By direct computation, we have

$$y_k^T d_{k+1} = -\frac{s_k^T y_k}{y_k^T s_k}(s_k^T g_{k+1}) = -v_k (s_k^T g_{k+1}) \quad (3.9)$$

Again, observe that the search direction (2.26) satisfies a variant of the Dai–Liao[22] conjugacy condition, where $v_k > 0$. This direction, after some algebra, can be written as follows:

$$d_{k+1} = -Q^{(3)}_{k+1} g_{k+1}$$

Where

$$Q^{(3)}_{k+1} = \frac{1}{y^T_{k-1} s_{k-1}} [s^T_{k-1} s_{k-1} I_n + (\theta_k y_{k-1} - s_{k-1})^T s_{k-1} - s^T_{k-1}(\theta_k y_{k-1} - s_{k-1})] \quad (3.10)$$

Observe that $Q^{(3)}_{k+1} = -\left(Q^{(3)}_{k+1}\right)^T$, Besides, $Q^{(3)}_{k+1}$ is a sum of a diagonal matrix and a skew symmetric one.

4. Convergence Analysis

We first state the following mild assumptions, which will be used in the proof of global convergence property.

Assumption (H).

(i) The level set $S = \{x : x \in R^n, f(x) \le f(x_1)\}$ is bounded, where x_1 is the starting point.

(ii) In a neighborhood Ω of S, f is continuously differentiable and its gradient g is Lipschitz continuously, namely, there exists a constant $L \ge 0$ such that

$$||g(x) - g(x_k)|| \le L||x - x_k||, \forall x, x_k \in \Omega \quad (4.1)$$

Under these assumptions on f there exists a constant $\Gamma \ge 0$, such that:

$$\|g(x)\| \le \Gamma, \forall x \in S \quad (4.2)$$

Although the search directions generated by (2.1), (2.10), (2.14) and (2.26) are always descent directions, to ensure convergence of the algorithm we need some Lemma.

Lemma 4.1.

Suppose that the assumption (H) (i) and(ii) hold. Then

$$\sum_{k=0}^{\infty} -\alpha_k g_k^T d_k < \infty \quad (4.3)$$

Proof:

from (1.17) and descent condition, we have

$$f_{k+1} - f_k \le \sigma \alpha_k g_k^T d_k$$

$$= -\sigma \alpha_k (\|g_{k+1}\|^2 + \left[1 + \frac{\|y_k\|^2}{y_k^T s_k}\right] \frac{(s_k^T g_{k+1})^2}{y_k^T s_k}) \le 0 \quad (4.4)$$

Therefore,{ f_k } is a decreasing sequence. Since f is bounded below, there exists a constant $f^{\wedge}$ such that

$$\lim_{k \to \infty} f_k = f^{\wedge} \quad (4.5)$$

it is follows that

$$\sum_{k=0}^{\infty} (f_k - f_{k+1}) = \lim_{\sigma \to \infty} \sum_{k=0}^{\sigma} (f_k - f_{k+1}) = \lim_{\sigma \to \infty} f_0 - f_{\sigma+1} = f_0 - f^{\wedge}$$

hence,

$$\sum_{k=0}^{\infty} (f_k - f_{k+1}) < +\infty \quad (4.6)$$

Lemma 4.2.

Suppose that the assumption (H) (i) and(ii), as well as the descent condition hold. Then

$$\sum_{k=0}^{\infty} \frac{(g_k^T d_k)^2}{\|d_k\|^2} < +\infty \quad (4.7)$$

Proof.

Subtracting $g_{k+1}^T d_{k+1}$ from both sides of (1.18) and using the Lipschitz condition, we get:

$$\alpha_k \geq \frac{(1-\delta)\left|g_{k+1}^T d_{k+1}\right|}{L\left\|d_{k+1}\right\|^2}$$

We know that for all k, $g_{k+1}^T d_{k+1} < 0$, therefore using **Lemma 4.1** we get immediately (4.7).

Proposition 4.3.

Suppose that assumption (H) (i) and (ii) hold, and consider any conjugate gradient algorithm (1.2), where d_{k+1} is a descent direction and α_k is obtained by the strong Wolfe line search (1.17) and (1.19). If

$$\sum_{k\geq 1} \frac{1}{\left\|d_{k+1}\right\|^2} = \infty, \qquad (4.8)$$

Then

$$\liminf_{k\to\infty} \left\|g_k\right\| = 0 \qquad (4.9)$$

For uniformly convex functions, we can prove that the norm of the direction d_{k+1}^{new} generated by (2.1), (2.10), (2.14) and (2.26) are bounded above. Therefore, by **Proposition 4.3**, we can prove the following result.

Theorem 4.4.

Suppose that assumption (H) (i) and (ii) hold, and consider the algorithm (1.2) and (2.1), (2.10), (2.14), where d_{k+1}^{new} is a descent direction and α_k is computed by the strong Wolfe line search (1.17) and (1.19). Suppose that *f* is a **uniformly convex function** on *S*, i.e. there exists a constant μ > 0 such that

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

(4.10)

For all $x,\ y \in N$ then

$$\lim_{k\to\infty} \left\|g_k\right\| = 0 \qquad (4.11)$$

Proof.

If the algorithm (1.2) and (2.1), (2.10), (2.14), from Lipschitz continuity, we have $\left\|y_k\right\| \leq L\left\|s_k\right\|$. On the other hand, from uniform convexity, $y_k^T s_k \geq \mu \left\|s_k\right\|^2$. Using the Cauchy inequality, assumption (H) (i) and (ii) and the above inequalities, we have

$$\left|\beta_k^{HS}\right| \leq \frac{\left|g_{k+1}^T y_k\right|}{\left|\frac{1}{\alpha_k} y_k^T s_k\right|} \leq \frac{L\Gamma}{\frac{1}{\alpha_k}\mu\left\|s_k\right\|} \leq \frac{\alpha_k L\Gamma}{\mu\left\|s_k\right\|}$$

(4.12)

$$\left|\varphi_k\right| \leq \frac{\left|s_k^T g_{k+1}\right|}{\left|y_k^T s_k\right|} \leq \frac{\Gamma}{\mu\left\|s_k\right\|}$$

(4.13)

$$\left|t_1\right| \leq 1 + \frac{\left\|y_k\right\|^2}{\left|y_k^T s_k\right|} \leq 1 + \frac{L^2}{\mu}$$

(4.14a)

$$\left|t_2\right| \leq 1 + 2\frac{\left\|y_k\right\|^2}{\left|y_k^T s_k\right|} \leq 1 + 2\frac{L^2}{\mu}$$

(4.14b)

$$\left|t_3\right| \leq \frac{2\left\|y_k\right\|^2}{\left|y_k^T s_k\right|} \leq 2\frac{L^2}{\mu}$$

(4.14c)

Therefore, using (4.12), (4.13) and (4.14)(a-c) in (2.1), we get

$$\left\|d_{k+1}\right\| = \left\|g_{k+1}\right\| + \left|\beta_k^{HS}\right|\left|\frac{1}{\alpha_k}\right|\left\|s_k\right\| + \left|\varphi_k\right|\left\|y_k\right\| + \left|\varphi_k\right|\left|t_1\right|\left\|s_k\right\| \leq \Gamma + \frac{\Gamma}{\mu}(2L + 1 + \frac{L^2}{\mu})$$

(4.15a)

$$\left\|d_{k+1}\right\| = \left\|g_{k+1}\right\| + \left|\beta_k^{HS}\right|\left|\frac{1}{\alpha_k}\right|\left\|s_k\right\| + \left|\varphi_k\right|\left\|y_k\right\| + \left|\varphi_k\right|\left|t_2\right|\left\|s_k\right\| \leq \Gamma + \frac{\Gamma}{\mu}(2L + 1 + 2\frac{L^2}{\mu})$$

(4.15b)

$$\|d_{k+1}\| = \|g_{k+1}\| + |\beta_k^{HS}| \left|\frac{1}{\alpha_k}\right| \|s_k\| + |\varphi_k| \|y_k\| + |\varphi_k| \|t_3\| \|s_k\| \leq \Gamma + \ldots$$

(4.15c)

showing that (4.8) is true. By **Proposition 4.3**, it follows that (4.9) is true, which for uniformly convex functions is equivalent to (4.11).

Theorem 4.5.

Suppose that assumption (H) (i) and (ii) hold, and consider the algorithm (1.2) and (2.26), where d_{k+1}^{new} is a sufficient descent direction and α_k is computed by the strong Wolfe line search (1.17) and (1.19). Suppose that *f* is a **uniformly convex function** on *S* (4.10), then $\lim_{k\to\infty}\|g_k\| = 0$

Proof.

If the algorithm (1.2) and (2.26), from Lipschitz continuity, we have $\|y_k\| \leq L\|s_k\|$. On the other hand, from uniform convexity, $y_k^T s_k \geq \mu\|s_k\|^2$. Using the Cauchy inequality, assumption (H) (i) and (ii) and the above inequalities, we have

$$|\theta_k| \leq \frac{|s_k^T s_k|}{|y_k^T s_k|} \leq \frac{\|s_k\|^2}{\mu\|s_k\|^2} \leq \frac{1}{\mu}$$

(4.16)

$$|\beta_k^{BM}| \leq \frac{|g_{k+1}^T(\theta_k y_k - s_k)|}{|y_k^T s_k|} \leq \frac{\Gamma(\frac{1}{\mu}L + 1)}{\mu\|s_k\|}$$

(4.17)

$$|\eta_k| \leq \frac{|s_k^T g_{k+1}|}{|y_k^T s_k|} \leq \frac{\Gamma}{\mu\|s_k\|}$$

(4.18)

Therefore, using (4.16), (4.17) and (4.18) in (2.26), we get

$$\|d_{k+1}\| = \|g_{k+1}\| + |\beta_k^{BM}| \|s_k\| + |\eta_k| \|\theta_k\| \|y_k\| + |\eta_k| \|s_k\| \leq \frac{\Gamma}{\mu}(2 + \ldots \leq \Gamma + \frac{\Gamma}{\mu}(2L + 2\frac{L^2}{\mu})$$

(4.19)

showing that (4.8) is true. By **Proposition 4.3**, it follows that (4.9) is true, which for uniformly convex functions is equivalent to (4.11).

5. Numerical Results.

The main work of this section is to report the performance of the new **three (1-3) TTHS & 4TT-CG** method on a set of test problems. The codes were written in Fortran and compiled with F77 (default compiler settings). All the tests were performed on a PC. We selected 34 large-scale unconstrained optimization test functions in generalized or extended form [5] (some from CUTE library). For each test function, we have taken10 numerical experiments with the number of variables n = 1000, 4500, 10000 and their details are given in the Appendix. In order to assess the reliability of our new proposed method, we have tested it against the standard Z1-CG method and Al-Bayati & Sharif (BS) [1] using the same test problems. The algorithm implements the acceleration Wolfe line search conditions with $\delta = 10^{-4}$, $\sigma = 0.8$ and the same stopping criterion $\|g_k\|_\infty \leq 10^{-6}$, where $\|\ .\ \|_\infty$ is the maximum absolute component of a vector. We also force these routines stopped if the iterations exceed 10000 or the number of function evaluations reach 15000 without achieving convergence. In Table **5.1** compares some numerical results for **1TTHS, 2TTHS** and **3TTHS** methods against **BS-CG** methods, Table **5.2** compares some numerical results for **1TTHS, 2TTHS** and **3TTHS** methods against **TTHS-CG** methods and in Table **5.3** compares the numerical results for **4TTHS** method against **BM-CG** methods; this tables 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 **5.4 and 5.5**, we have compared the percentage performance of the **1TTHS, 2TTHS** and **3TTHS** against the **BS-CG** method at the first and in the second table we compared this methods against the **TTHS-CG** method, in the final Table **5.6** we have compared the percentage performance of the **4TT-CG** method against the scaled CG method of **BM-CG** method; all of this compared with respect to NOI; NOFG and TIME taking over all the tools as 100%.

Table 1:COMPARISON BETWEEN THE NEW 1TTHS, 2TTHS AND 3TTHS AND BS METHODS FOR THE TOTAL OF (34) PROBLEMS WITH n= 1000, 4500, 10000

Prob.	1TTHS-CG			2TTHS-CG			3TTHS-CG			BS-CG /2010		
	NOI	NOFG	TIME	NOI	NOFG	TIME	NOI	NOFG	TIME	NOI	NOFG	TIME
1	172	522	0.81	181	558	0.88	155	476	0.72	97	172	0.26
2	391	1105	0.19	482	1648	0.2	389	1106	0.17	84	182	0.05
3	46	172	0.01	49	170	0.03	49	176	0.03	199	4841	0.59
4	9	30	0.02	9	30	0.03	9	30	0.03	12	27	0.04
5	97	368	0.07	96	325	0.08	96	419	0.06	143	1903	0.45
6	84	276	0.31	84	227	0.21	84	251	0.27	58	488	1.22
7	384	1075	0.17	410	1177	0.17	488	1366	0.21	97	187	0.03
8	6	21	0.05	6	21	0.04	6	21	0.07	12	27	0.09
9	33	102	0	33	102	0.01	33	102	0.02	42	81	0.01
10	27	84	0.07	24	75	0.04	24	75	0.06	27	57	0.05
11	85	262	0.17	77	257	0.14	111	359	0.22	164	260	0.25
12	5	16	0.02	5	16	0.02	5	16	0	25	524	0.33
13	21	63	0.01	21	63	0.02	21	63	0.03	18	36	0.02
14	21	67	0.01	21	67	0.03	21	67	0.02	32	55	0.02
15	24	73	0.03	24	73	0.03	24	73	0.04	43	75	0.03
16	251	735	0.22	225	863	0.25	242	705	0.18	250	397	0.14
17	117	478	0.11	123	470	0.11	122	516	0.13	208	3619	0.65
18	29	90	0.06	29	90	0.06	29	90	0.06	52	147	0.03
19	9	30	0.04	9	30	0.05	9	30	0.03	12	27	0.04
20	109	610	0.11	110	480	0.08	118	676	0.12	484	12566	2.48
21	99	276	0.16	99	276	0.18	99	276	0.17	61	99	0.08
22	26	78	0.02	26	78	0.02	26	78	0.02	24	51	0.01
23	44	125	0.02	46	135	0.03	46	135	0.03	66	113	0.05
24	144	418	0.26	113	330	0.22	122	356	0.25	69	121	0.09
25	27	96	0.02	27	96	0.02	27	96	0.01	28	71	0.01

26	12	42	0.05	12	42	0.02	12	42	0.02	12	33	0.04
27	9	30	0.03	9	30	0.03	9	30	0.03	9	21	0.05
28	6	21	0	6	21	0.03	6	21	0.03	9	21	0.04
29	27	84	0.06	24	75	0.06	24	75	0.06	27	57	0.06
30	162	469	0.09	172	498	0.08	172	497	0.09	130	227	0.05
31	5	38	0.02	5	38	0.02	5	38	0.01	10	88	0.01
32	82	230	0.05	76	214	0.04	76	214	0.07	208	1769	0.39
33	18	42	0.03	18	42	0.03	18	42	0.03	24	30	0.04
34	36	102	0.02	36	102	0.01	36	102	0.02	39	75	0.02
Total	2617	8230	3.31	2687	8719	3.27	2713	8619	3.31	2775	28447	7.72

Table 2:COMPARISON BETWEEN THE NEW 1TTHS, 2TTHS AND 3TTHS AND TTHS METHODS FOR THE TOTAL OF (34) PROBLEMS WITH n= 1000, 4500, 10000

Prob.	1TTHS-CG			2TTHS-CG			3TTHS-CG			TTHS-CG / 2006		
	NOI	NOFG	TIME	NOI	NOFG	TIME	NOI	NOFG	TIME	NOI	NOFG	TIME
1	172	522	0.81	181	558	0.88	155	476	0.72	98	167	0.25
2	391	1105	0.19	482	1648	0.2	389	1106	0.17	10081	10159	0.62
3	46	172	0.01	49	170	0.03	49	176	0.03	52	137	0.03
4	9	30	0.02	9	30	0.03	9	30	0.03	12	27	0.05
5	97	368	0.07	96	325	0.08	96	419	0.06	136	1621	0.33
6	84	276	0.31	84	227	0.21	84	251	0.27	43	79	0.11
7	384	1075	0.17	410	1177	0.17	488	1366	0.21	118	209	0.05
8	6	21	0.05	6	21	0.04	6	21	0.07	12	27	0.08
9	33	102	0	33	102	0.01	33	102	0.02	42	81	0.01
10	27	84	0.07	24	75	0.04	24	75	0.06	27	57	0.06
11	85	262	0.17	77	257	0.14	111	359	0.22	164	260	0.25
12	5	16	0.02	5	16	0.02	5	16	0	25	524	0.32
13	21	63	0.01	21	63	0.02	21	63	0.03	18	36	0.03
14	21	67	0.01	21	67	0.03	21	67	0.02	32	55	0.03
15	24	73	0.03	24	73	0.03	24	73	0.04	43	75	0.02
16	251	735	0.22	225	863	0.25	242	705	0.18	238	397	0.14
17	117	478	0.11	123	470	0.11	122	516	0.13	181	2673	0.63
18	29	90	0.06	29	90	0.06	29	90	0.06	52	147	0.01
19	9	30	0.04	9	30	0.05	9	30	0.03	12	27	0.05
20	109	610	0.11	110	480	0.08	118	676	0.12	439	11497	2.01

21	99	276	0.16	99	276	0.18	99	276	0.17	56	91	0.08
22	26	78	0.02	26	78	0.02	26	78	0.02	24	51	0
23	44	125	0.02	46	135	0.03	46	135	0.03	63	109	0.01
24	144	418	0.26	113	330	0.22	122	356	0.25	65	114	0.1
25	27	96	0.02	27	96	0.02	27	96	0.01	28	71	0.02
26	12	42	0.05	12	42	0.02	12	42	0.02	12	33	0.06
27	9	30	0.03	9	30	0.03	9	30	0.03	9	21	0.03
28	6	21	0	6	21	0.03	6	21	0.03	9	21	0.03
29	27	84	0.06	24	75	0.06	24	75	0.06	27	57	0.06
30	162	469	0.09	172	498	0.08	172	497	0.09	122	209	0.04
31	5	38	0.02	5	38	0.02	5	38	0.01	10	88	0.03
32	82	230	0.05	76	214	0.04	76	214	0.07	184	1558	0.33
33	18	42	0.03	18	42	0.03	18	42	0.03	24	30	0.01
34	36	102	0.02	36	102	0.01	36	102	0.02	39	75	0.02
Total	2617	8230	3.31	2687	8719	3.27	2713	8619	3.31	12497	30783	5.9

Table 3: COMPARISON BETWEEN THE NEW 4TTCG AND BM-CG METHODS FOR THE TOTAL OF (34) PROBLEMS WITH n= 1000, 4500, 10000

Prob.	**4TT-CG**			**BM-CG**		
	NOI	**NOFG**	**TIME**	**NOI**	**NOFG**	**TIME**
1	134	354	0.6	98	172	0.27
2	37	107	0.09	36	71	0.01
3	29	101	0.09	57	135	0.02
4	9	30	0.09	12	27	0.05
5	63	211	0.22	97	982	0.2
6	18	51	0.44	35	60	0.09
7	9	27	0.02	14	27	0
8	6	21	0.25	12	27	0.09
9	17	54	0.03	57	93	0.02
10	21	69	0.31	20	43	0.05
11	34	108	0.29	164	260	0.23
12	5	16	0.01	25	524	0.33
13	15	45	0.13	18	36	0.02
14	17	55	0.16	32	55	0.03
15	18	55	0.17	43	75	0.02
16	165	438	0.52	179	290	0.11

17	65	241	0.27	145	1807	0.46
18	21	63	0.35	52	147	0.01
19	9	30	0.11	12	27	0.04
20	58	259	0.21	589	16912	3.76
21	24	72	0.25	27	52	0.05
22	18	57	0.01	16	35	0
23	21	66	0.07	55	97	0.01
24	24	72	0.26	35	67	0.07
25	22	83	0.07	20	55	0.02
26	12	42	0.19	12	33	0.08
27	9	30	0.19	9	21	0.01
28	6	21	0.12	9	21	0.03
29	21	69	0.29	20	43	0.05
30	87	230	0.2	93	157	0.03
31	5	38	0.06	10	88	0.04
32	66	205	0.2	162	1421	0.38
33	18	42	0.11	24	30	0.03
34	18	57	0.04	18	39	0.01
Total	**1101**	**3419**	**6.42**	**2207**	**23929**	**6.62**

Table 4: PERCENTAGE PERFORMANCE OF TABLE 1

Tools	**TTHS-CG Method**	**1TTHS-CG Method**	**2TTHS-CG Method**	**3TTHS-CG Method**
NOI	100%	94.3%	96.8%	97.7%
NOFG	100%	28.9%	30.6%	30.2%
CPU	100%	42.8%	42.3%	42.8%

Table 5: PERCENTAGE PERFORMANCE OF TABLE 2

Tools	**BS-CG Method**	**1TTHS-CG Method**	**2TTHS-CG Method**	**3TTHS-CG Method**
NOI	100%	20.9%	21.5%	21.7%
NOFG	100%	26.7%	28.3%	27.9%
CPU	100%	56.1%	55.4%	56.1%

Table 6: PERCENTAGE PERFORMANCE OF TABLE 3

Tools	BM Method	4TT-CG Method
NOI	100%	49.8%
NOFG	100%	14.2%
CPU	100%	96.9%

6. Discussion.

It is clear from **Table 4** that taking, over all, the tools as a 100% for the **BS-CG** method, the three New three-term methods has an improvement, in about New **1TTHS-CG** method (5.7%) NOI , (71.1%) NOFG and (57.2%) TIME, in about New **2TTHS-CG** method (3.2%) NOI , (69.4%) NOFG and (57.7%) TIME; In about New **3TTHS-CG** method (2.3%) NOI , (69.8%) NOFG and (57.2%) TIME.

In from **Table 5** that taking, over all, the tools for **TTHS-CG** method, the three New three-term methods has an improvement, in about New **1TTHS-CG** method (79.1%) NOI , (73.3%) NOFG and (43.9%) TIME, in about New **2TTHS-CG** method (78.5%) NOI, (71.7%) NOFG and (44.6%) TIME; In about New **3TTHS-CG** method (78.3%) NOI , (72.1%) NOFG and (43.9%) TIME.

It is clear from **Table 6** that taking, over all, the tools for **BM-CG** method, the fourth New three-term method (**4TT-CG**) has an improvement, in (50.2%) NOI , (85.8%) NOFG and (3.1%) TIME. These results indicate that the four new three term method is in general is the best**.**

Appendix

1)Trigonometric **2**) Extended Beale **3**) Penalty **4**) Raydan 2 **5**) Generalized Tridiagonal1 **6**) Extended Three Expo-Terms **7**) Diagonal 4 **8**) Diagonal 5 **9**) Extended Himmelblau **10**) Extended PSC1 **11**) Extended BD1 **12**) Extended EP1 **13**) DIXMAANA (CUTE) **14**) DIXMAANB (CUTE) **15**) DIXMAANC (CUTE) **16**) Broyden Tridiagonal **17**) EDENSCH (CUTE) **18**) HIMMELBHA **19**) DIAGONAL 6 **20**) ENGVAL1 (CUTE) **21**) DENSCHNA (CUTE) **22**) DENSCHNB (CUTE) **23**) DENSCHNF (CUTE) **24**) Extended Block-Diagonal BD2 **25**) Generalized quartic GQ1 **26**) DIAGONAL 7 **27**) DIAGONAL 8 **28**) Full Hessian **29**) SINCOS **30**) Generalized quartic GQ2 **31**) ARGLINB (CUTE) **32**) FLETCHCR (CUTE) **33**) HIMMELBG (CUTE) **34**) HIMMELBH (CUTE).

References

[1] A.Y. Al-Bayati, W.H. Sharif. A new three-term conjugate gradient method for unconstrained optimization, Can. J. Sci. Eng. Math. 1 (5) (2010), 108–124.

[2] D.C. Liu, J. Nocedal. On the limited memory BFGS method for large scale optimization, Mathematical Program. 45 (1989) , 503–528.

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

[4] E. Polak, G. Ribière. Note Sur la Convergence de Directions Conjuguée, Rev. Francaise Informat Recherche Operationelle, 3e Année, 16 (1969), 35–43.

[5] E.D. Dolan, J.J. Moré, Benchmarking optimization software with performance profiles, Math. Program. 91 (2002), 201–213.

[6] E.M.L. Beale. A derivative of conjugate gradients, in: F.A. Lootsma (Ed.), Numerical Methods for Nonlinear Optimization, Academic Press, London, (1972), 39–43.

[7] J. Nocedal. Conjugate gradient methods and nonlinear optimization, In: L. Adams, J.L. Nazareth, (eds.) (Linear and Nonlinear Conjugate Gradient—Related Methods) ,(1996), 9–23. SIAM, Philadelphia.

[8] J. Zhang, Y. Xiao, Z. Wei. Nonlinear conjugate gradient methods with sufficient descent condition for large-scale unconstrained optimization, Math. Prob. Eng. (2009), Article ID 243290, 16. DOI: 10.1155/2009/243290.

[9] J.C. Gilbert, J. Nocedal. Global convergence properties of conjugate gradient methods for optimization, SIAM J. Optim. 2 (1992), 21–42.

[10] L. Grippo, S. Lucidi. A globally convergent version of the Polak–Ribiére conjugate gradient method, Math. Program. 78 (1997), 375–391.

[11] L. Nazareth. A conjugate direction algorithm without line search, Journal of Optimization Theory and Applications 23 (1977) 373–387.

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

[13] L. Zhang, W.J. Zhou, D.H. Li. Some descent three-term conjugate gradient methods and their global convergence. Optimization Methods and Software , 22 (2007), 697– 711.

[14] M. Al-Baali, Descent property and global convergence of the Fletcher–Reeves method with inexact line search, IMA J. Numer. Anal. 5 (1985), 121–124.

[15] M.R. Hestenes, E. Stiefel. Methods of conjugate gradient for solving linear system.Journal of Research of the National Bureau of Standards , 49 (1952), 409–436.

[16] N. Andrei, Acceleration of conjugate gradient algorithms for unconstrained optimization, Applied Mathematical Computational. 213 (2009), 361–369.

[17] N. Andrei. Performance profiles of conjugate gradient algorithms for unconstrained optimization, In: Floudas, C.A., Pardalos, P.M. (eds.) Encyclopedia of Optimization, 2nd edition. (2009), 2938–2953. Springer, New York.

[18] R. Fletcher, C. Reeves. Function minimization by conjugate gradients, Comput. Journal, 7 (1964), 149–154.

[19] R. Fletcher. Practical Methods of Optimization, vol. I: Unconstrained O ptimization ,2nd edition. New York, Wiley, (1987).

[20] Y. Hu, C. Storey. Global convergence results for conjugate gradient methods, J. Optim. Theory Appl. 71 (1991), 399–405.

[21] Y.H. Dai, Y.X. Yuan. A nonlinear conjugate gradient method with a strong global convergence property. SIAM Journal on Optimization, 10 (1999), 177–182.

[22] Y.-H. Dai and L.-Z. Liao, "New conjugacy conditions and related nonlinear conjugate gradient methods," Applied Mathematics and Optimization, vol. 43, no. 1, pp. 87–101, 2001.

[23] Al-Bayati, A. Y., (1991), A new family of self-scaling variable metric algorithms of unconstrained optimization, J. of Education and Science, University of Mosul, Vol. 12, pp. 25-54.