American Journal of Economics and Business Administration 1 (2) 97-104, 2009
ISSN 1945-5488

Two-Versions of Conjugate Gradient-Algorithms Based on Conjugacy Conditions for Unconstrained Optimization

Abbas Y. AL-Bayati, A.J. Salim and Khalel K. Abbo
Department of Mathematics, Mosul University, Mosul, Iraq

Abstract: Problem statement: (CG) algorithms, which we had investigated in this study, were widely used in optimization, especially for large scale optimization problems, because it did not need the storage of any matrix. The purpose of this construction was to find new CG-algorithms suitable for solving large scale optimization problems. **Approach:** Based on pure conjugacy condition and quadratic convex function two new versions of (CG) algorithms were derived and observed that they were generate descent directions for each iteration, the global convergence analysis of these algorithms with Wolfe line search conditions had been proved. **Results:** Numerical results for some standard test functions were reported and compared with the classical Fletcher-Reeves and Hestenes-Stiefel algorithms showing considerable improving over these standard CG-algorithms. **Conclusion:** Two new versions of CG-algorithms were proposed in this study with their numerical properties and convergence analysis and they were out perform on the standard HS and FR CG-algorithms.

Key words: (CG) algorithms, exact line searches, global convergence properties

INTRODUCTION

The problem of interest can be stated as finding a local x* to the unconstrained optimization problem:

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

where, f: $R^n \to R$ is continuously differentiable and its gradient is available and denoted by g. There are different type of iteration algorithms for solving the problem given in (1); all these algorithms uses the iteration of the form:

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

Where:
X_0 = Starting point and
α_k = Step-size computed by line search procedure
d_k = A descent direction

If $f \in C^2$ and the Hessian matrix $G = \nabla^2 f(x)$ is available and positive definite then an ideal choice for d_k is the Newton direction[7] given by:

$$d_{k+1} = -G^{-1} g_{k+1} \tag{3}$$

It is shown that when G_k is positive definite and x_k is lies in some neighborhood of x* then the sequence $\{x_k\}$ generated by (2) and (3) converges and order of convergence is second order. These local convergence properties represent the ideal local behavior which other algorithms aim to emulate as far as possible[6], in spite of these desirable properties of Newton's algorithm also it has some drawbacks such as dealing with n × n matrix and when x_k is remote from x* the algorithm may not defined when G_k is not positive. Therefore, other algorithms can be used for solving the problem (1) such as Quasi-Newton algorithms which are modifications of Newton's algorithm and uses direction of the form:

$$d_{k+1} = -H_{k+1} g_{k+1} \tag{4}$$

where, H_{k+1} is an approximation of the inverse Hessian matrix. The Conjugate Gradient (CG) algorithm is suitable approach to solve the minimization problem given in (1) when n is large. If the (CG) algorithm is used to minimize non-quadratic objective functions the related algorithm is called the non-linear (CG) algorithm[12,14,15]. The search directions for CG-algorithm has the following form:

$$d_{k+1} = -g_{k+1} + \beta_k d_k \tag{5}$$

Here β_k is a scalar known as the (CG) parameter. Different CG-algorithms correspond to different

Corresponding Author: Abbas Y. AL-Bayati, Department of Mathematics, Mosul University, Mosul, Iraq

choices for the parameter β_k, therefore a crucial element in any (CG) algorithm is the definition of formula β_k; some well-known (CG) algorithms include the Hestenes-Stiefel (HS) algorithm, the Fletcher-Reeves (FR), the Polak-Ribiere (PR) and the Dai-Yuan (DY) algorithms[4,9,10,11]:

$$\beta_k^{HS} = \frac{g_{k+1}^T y_k}{d_k^T y_k} \tag{6}$$

$$\beta_k^{FR} = \frac{g_{k+1}^T g_{k+1}}{g_k^T g_k} \tag{7}$$

$$\beta_k^{PR} = \frac{g_{k+1}^T y_k}{g_{k+1}^T g_k} \tag{8}$$

$$\beta_k^{DY} = \frac{g_{k+1}^T g_{k+1}}{d_k^T y_k} \tag{9}$$

where, $y_k = g_{k+1}-g_k$.

Hager and Zhang[8] shown that CG-algorithms with $g_{k+1}^T g_{k+1}$ in the numerator of β_k has strong global convergence properties with exact and inexact line searches especially with the following Wolfe conditions:

$$f(x_k + \alpha_k d_k) \le f(x_k) + c_1 \alpha_k g_k^T d_k \tag{10a}$$

$$g(x_k + \alpha_k d_k)^T d_k \ge c_2 g_k{}^T d_k \tag{10b}$$

where, $0<c_1<c_2<1$. For some CG-algorithms stronger version of Wolfe conditions i.e., (10a) and:

$$\left| g_{k+1}{}^T d_k \right| \le - c_2 g_k{}^T d_k \tag{11}$$

Are need to ensure global convergence and hence stability[8]. But these algorithms has poor performance in practice. On the other hand the CG-algorithms with $g_{k+1}^T y_k$ in the numerator has uncertain global convergence for general non-linear functions but has good performance in practice. One of the open questions in optimization is whether can we construct a (CG) that has both global convergence and good numerical performance in practical computation? In this study we try to derive new CG-algorithms with global convergence property and acceptable performance in practice. All the algorithms mentioned earlier (Newton algorithm, Quasi-Newton algorithm and CG-algorithms) are called conjugate direction algorithms since they generates a conjugate directions i.e., the search directions generated by these algorithms satisfies the following equation:

$$d_i{}^T G d_j = 0 \ \forall \ i \ne j \tag{12}$$

When the objective function is quadratic and convex function and step size α_k is exact. For general non linear functions, we know by the mean value theorem that there exists some $\gamma \in (0, 1)$ such that:

$$\alpha_k^{-1} d_{k+1}{}^T y_k = d_{k+1}{}^T \nabla^2 f(x_k + \gamma \alpha_k d_k)\, d_k$$

Therefore, it is reasonable[5] to replace (12) with the following conjugacy condition:

$$d_{k+1}^T y_k = 0 \tag{13}$$

Which is called pure conjugacy condition.

Dai and Liao[5] combined the search direction given in (3) or (4) with secant equation to modify the pure conjugacy condition (13) as follows:

$$d_{k+1} = -B_{k+1} g_{k+1} \tag{14}$$

where, B_{k+1} symmetric positive $n \times n$ matrix, satisfying the quasi-Newton or (secant) equation:

$$B_{k+1} y_k = s_k \tag{15}$$

where, $s_k = x_{k+1} - x_k$, therefore:

$$\begin{aligned} d_{k+1}^T y_k &= -(B_{k+1} g_{k+1})^T y_k \\ &= -g_{k+1}^T (B_{k+1} y_k) = -g_{k+1}^T s_k \end{aligned} \tag{16}$$

This relation shows that (13) hold if the line search is exact i.e., $g_{k+1}{}^T s_k = 0$. However practical numerical algorithms normally adopt inexact line searches. For this reason, it seems more reasonable to replace the conjugacy condition (13) with the condition:

$$d_k^T y_k = -t g_{k+1}^T s_k \tag{17}$$

where, $t>0$ is a scalar. The main object of this study is to find new (CG) algorithms with new search directions d_k having the same form of (5). This is done in materials and algorithms as well as the descent property and global convergent property will be proved and numerical results will be reported and compared with some standard CG algorithms.

MATERIALS AND ALGORITHMS

Derivation of two new versions of CG-algorithms: It is known that all conjugate direction algorithms generate conjugate directions at least theoretically[6] and hence the key element for derivation of the new algorithms is the pure conjugacy condition (13), also in derivation of all conjugate direction algorithms it is assumed that the objective function is convex and quadratic, therefore we begin with convex quadratic function q(x) defined by:

$$q(x) = \frac{1}{2} x^T G x \tag{18}$$

where, $x \in R^n$ and G is positive definite $n \times n$ matrix. Since q(x) is strictly convex then G is diagonal and gradient of q(x) is given by:

$$\nabla q(x) = Gx \tag{19}$$

The main property of quadratic function is:

$$g_{k+1} - g_k = G(x_{k+1} - x_k)$$

or

$$y_k = G s_k \tag{20}$$

From (18-20) we can write:

$$G^{-1} = \frac{s_k{}^T y_k}{y_k{}^T y_k}\ I_{n\times n} \tag{21a}$$

Therefore Newton direction (3) for function defined in (18) can be written as:

$$d^N_{k+1} = -\left(\frac{s_k^T y_K}{y_k^T y_K}\right) g_{k+1} \tag{21b}$$

Use the conjugacy condition (13) because Newton directions are conjugate with exact line searches:

$$d^T_{k+1}\ y_k = -\left(\frac{s_k^T y_k}{y_k^T y_K}\right) g^T_{k+1} y_k = 0 \tag{22}$$

Similarly CG algorithms generates conjugate:

$$d^{CG}_{k+1}{}^T y_k = -\ g_{k+1}{}^T y_k + \beta_k d_k{}^T y_k = 0 \tag{23}$$

Use (22) and (23) to get:

$$\beta_k = (1 - \frac{s_k{}^T y_k}{y_k{}^T y_k}) \frac{g^T_{k+1} y_k}{d^T_k y_k} \tag{24}$$

$$d_{k+1} = -g_{k+1}\ +\ (1 - \frac{s_k{}^T y_k}{y_k{}^T y_k}) \frac{g^T_{k+1} y_k}{d^T_k y_k}\ d_k$$

Put $s_k = \alpha_k d_k$

$$d_{k+1} = -g_{k+1}\ +\ (1 - \frac{s_k{}^T y_k}{y_k{}^T y_k}) \frac{g^T_{k+1} y_k}{s^T_k y_k}\ s_k \tag{25}$$

Where:

$$\beta_k{}^{v1} = (1 - \frac{s_k{}^T y_k}{y_k{}^T y_k}) \frac{g^T_{k+1} y_k}{s^T_k y_k} \tag{26}$$

We can therefore modify the Eq. 25 and 26 by using the idea of Dai and Laio[5] and combining the quasi-Newton condition with pure conjugacy condition:

$$\begin{aligned} d_{k+1}{}^T y_k &= -(G^{-1} g_{k+1})^T y_k \\ &= -g_{k+1}{}^T G^{-1} y_k = -g_{k+1}{}^T s_k \end{aligned} \tag{27}$$

or

$$-\frac{s_k{}^T y_k}{y_k{}^T y_k}\ g_{k+1}{}^T y_k + g_{k+1}{}^T s_k = 0 \tag{28}$$

and

$$d_{k+1}{}^T y_k = -g_{k+1}{}^T y_k + \beta_k d_k{}^T y_k = 0 \tag{29}$$

From (28) and (29) we get:

$$\beta_k{}^{v2}\ = (1 - \frac{s_k^T y_k}{y_k^T yk}) \frac{g^T_{k+1} y_k}{d^T_k y_k} + \frac{s^T_k g_{k+1}}{d^T_k y_k} \tag{30}$$

And letting $s_k = \alpha_k d_k$:

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

It seems from (30) if exact line search used i.e $s_k{}^T g_{k+1} = 0$ then (30) reduces to (26).

For convenience, we summarize the above algorithms as the following algorithms.

Algorithm 1: (New V1 Algorithm): The search direction of this new algorithm is defined as:

$$d = -g_{k+1} + \beta_k^{v1} \ d_k$$

when β^{v1} computed as in (26). If Powell restart satisfied then $d_{k+1} = -g_{k+1}$ else d_{k+1} = d and compute initial $\alpha_k = \alpha_{k-1}\frac{\|d_k\|}{\|d_{k-1}\|}$ go to step (2).

Similarly we can summarize the new v2 algorithm as.

Algorithm 2: (New V2 Algorithm): The search direction of this new algorithm is defined as:

$$d = -g_{k+1} + \beta_k^{v2} \ d_k$$

when β^{v2} computed as in (30). If Powell restart satisfied then $d_{k+1} = g_{k+1}$ else d_{k+1} =d and compute initial $\alpha_k = \alpha_{k-1}\frac{\|d_k\|}{\|d_{k-1}\|}$ go to step (2).

Convergence analysis: In the investigated of the global convergence analysis of many iteration algorithms, the following assumption is often needed.

Assumption 1:

- The level set S = $\{x \in R^n$: $f(x) \le f(x_0)$ is bounded.
- In some neighborhood N of S f is continuously differentiable and its gradient is lipschitz continuous i.e., $\exists$ a constant L>0 s.t:

$$\|g(x) - g(y)\| \quad \le \quad L \quad \|x - y\| \forall x, \ y \in N \qquad (31)$$

Note that assumption A implies that $\exists$ a constant γ>0 such that:

$$\|g_k\| \le \gamma \ \forall \ \ x_k \in \ N \qquad (32)$$

In order to ensure global convergence of our algorithms we need to compute the step size α_k. The Wolfe line search consists of finding α_k satisfying (10a) and (10b). The following lemma, called the Zoutendijk condition is often used to prove global convergence of (CG) algorithms. It was originally given by Zoutendijk[16] and Wolfe[13].

Lemma 1: Let the assumption 1 holds, the sequence $\{x_k\}$ be generated by (2) and (5) and d_k is descent direction $\forall$ k i.e $g_k^T d_k < 0$. If α_k satisfies the Wolfe conditions (10a) and (10 b) or strong Wolfe conditions (10a) and (11) then we have:

$$\sum_{k=0}^{\infty} \frac{\left(g_k^T d_K\right)^2}{\|d_K\|^2} < +\infty \qquad (33)$$

for proof[16] or[13].

In the investigated of the global convergence analysis for many CG-algorithms. The descent or sufficient descent condition plays an important role. In the following theorem we show that the new v1 algorithm produces sufficient descent directions i.e., $g_k^T d_k \le -c\|g_k\|$ where c is some positive scalar.

Theorem 1: If the assumption 1 holds and α_k satisfies the Wolfe conditions then the search directions generated by (25) are descent directions $\forall$k.

Proof: For initial direction (k = 0) we have:

$$d_1 = -g_1 \ \rightarrow g_1^T d_1 = -\|g_1\|^2 \ < \ 0$$

Suppose $g_j^T d_j \ < \ 0 \qquad \forall \ j = 1....k:$

$$g_{k+1}^T d_{k+1} = -g_{k+1}^T g_{k+1} + \left(1 - \frac{s_k^T y_k}{y_\circ^T y_k}\right)\frac{g_{k+1}^T y_k}{s_k^T y_k} g_{k+1}^T s_k$$

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

From lipschitz condition $y_k^T g_{k+1} \ \le \ L \ s_k^T g_{k+1}$.

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

Again form Lipschitz condition:

$$y_k^T y_k \le L \ y_k^T s_k \rightarrow \frac{L}{y_k^T y_k} \ge \frac{1}{y_k^T s_k}$$

Therefore:

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

Note that from Wolfe conditions $s_k^T y_k \ > \ 0$ then we have two cases either L≥1 or 0< L<1, if L≥1 then d_k is a descent direction for all k. On the other hand if 0<L<1 then we have:

$$s_k^T y_k = \ s_k^T g_{k+1} - s_k^T g_k > s_k^T g_{k+1} \ \ \text{since } s_k^T g_k < 0$$

Also:

$$g_{k+1}{}^T d_{k+1} \le -g_{k+1}{}^T g_{k+1} + L\, s_k{}^T g_{k+1} - L^2 s_k{}^T g_{k+1}$$

If:

$$s_k{}^T g_{k+1} < 0$$

then:

$$g_{k+1}{}^T d_{k+1} < 0$$

Since 0<L<1 and if $s_k{}^T g_{k+1} > 0$ then:

$$L\, s_k{}^T g_{k+1} < g_{k+1}{}^T g_{k+1} + L^2\, s_k{}^T g_{k+1}$$

and hence $d_{k+1}{}^T g_{k+1} < 0$; so the proof of the Theorem 1 is completed.

Theorem 2: The global convergence of the new v1: Consider the iterative procedure $x_{k+1} = x_k + \alpha_k d_k$ where d_k is defined by (25) and suppose that the assumption 1 holds. If α_k satisfies the Wolfe conditions (10a) and (10b) then Algorithm 1 either stops at stationary point i.e., $\|g_k\| = 0$ or $\lim_{k\to\infty} \inf \|g_k\| = 0$.

Proof: The proof of theorem (2) is by contradiction i.e., if theorem (2) is not true then $\|g_k\| \neq 0$ then there exists a positive scalar $\gamma>0$ such that:

$$\|g_k\| > \gamma,\ \forall k$$

then:

$$d_{k+1}{}^T g_{k+1} = -g_{k+1}{}^T g_{k+1} + \frac{y_k^T g_{k+1}}{s_k^T y_k} s_k{}^T g_{k+1} - \frac{y_k^T g_{k+1}}{y_k^T y_k} s_k{}^T g_{k+1} \quad (34)$$

Use second Wolfe condition (10b) and Lipschifz condition for $y_k{}^T y_y \le L\ \ s_k{}^T y_k$. Therefore:

$$d_{k+1}{}^T g_{k+1} \ge -g_{k+1}{}^T g_{k+1} + c_2\, L\, \frac{y_k^T g_{k+1}}{y_k^T y_k} s_k{}^T g_k$$
$$-c_2 \frac{y_k^T g_{k+1}}{y_k^T y_k} s_k{}^T g_k$$

Note that:

$$\begin{aligned} y_k^T y_k &= g_{k+1}^T g_{k+1} - 2 g_k^T g_{k+1} + g_k^T g_k \\ &\ge g_{k+1}^T g_{k+1} - g_k^T g_{k+1} = y_k^T g_{k+1} \end{aligned}$$

$$d_{k+1}^T g_{k+1} \ge -\|g_{k+1}\|^2 + \lambda g_k^T s_k \quad (35)$$

When $\gamma = c_2(L-1)$; Divide both sides of (35) by $\|g_{k+1}\|^2$ and take the squares to get:

$$\frac{1}{\lambda^2}\left(\frac{d_{k+1}^T g_{k+1}}{\|g_{k+1}\|^2} + 1\right)^2 \ge \frac{\|s_k\|^2 \|g_k\|^2 \cos^2\theta_k}{\|g_{k+1}\|^2}$$

Since:

$$\left(g_k{}^T s_k\right)^2 = \|s_k\|^2 \|g_k\|^2 \cos^2\theta_k$$

then:

$$\frac{1}{\lambda^2 \|s_k\|^2}\left(\frac{d_{k+1}{}^T g_{k+1}}{\|g_{k+1}\|^2} + 1\right)^2 \ge \|g_k\|^2 \cos^2\theta_k \ge \gamma^2 \cos^2\theta_k$$

Taking the summation of the above equality from k = o to k = ∞ yields:

$$\frac{1}{\lambda^2 \|s_k\|^2} \sum_{k=0}^{\infty} \left(\frac{d_{k+1}^T g_{k+1}}{\|g_{k+1}\|^2} + 1\right)^2 \ge \sum_{k=0}^{\infty} \frac{\left(g_k^T s_k\right)^2}{\|s_k\|^2} \ge \sum_{k=0}^{\infty} \gamma^2 \cos^2 = \infty$$

Contradiction with Zountendijk theorem. Therefore $\|g_k\| = 0$.

Note: To study the convergence analysis of the new Algorithm 2 we will give only the conditions for descent property since the algorithm is not in general generates descent directions except under suitable conditions.

Theorem 3: If the gradient of the objective function is Lipschitz continuous with L>0 and if:

$$L \frac{\left(s_k^T g_{k+1}\right)^2}{s_k^T y_k} \le \frac{3}{4} g_{k+1}{}^T g_{k+1} \quad (36)$$

Then the search directions generated by Algorithm 2 are descent directions.

Proof: For initial direction $d_1 = -g_1 \ \to d_1{}^T g_1 = -\|g_1\|^2 < 0$ now let $g_j{}^T d_j < 0$, $j = 1 \dots k\text{-}1$ assuming (36) holds for ∀ j then:

$$d_{k+1}{}^T g_{k+1} = -g_{k+1}{}^T g_{k+1} + \beta_k d_k{}^T g_{k+1}$$

It is clear that for exact line searches the directions are descent for all k. We assume that the parameter α_k satisfies Wolfe conditions therefore:

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

Therefore using Lipschitz condition:

$$\frac{g_{k+1}^T y_k}{y_k^T y_k} \le \frac{g_{k+1}^T s_k}{y_k^T s_k}$$

Then:

$$d_{k+1}{}^T g_{k+1} \le -g_{k+1}{}^T g_{k+1} + \frac{g_{k+1}^T y_k}{s_k^T y_k} s_k{}^T g_{k+1} \tag{37}$$

But:

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

Use the fact $u^T v \le \frac{1}{2}(u^2 + v^2)$ with $u = \frac{\left(s_k^T y_k\right)}{\sqrt{2}} g_{k+1}$ and $v = \sqrt{2}(s_k{}^T g_{k+1})\ y_k$ to get:

$$\begin{aligned} \frac{\left(g_{k+1}^T y_k\right)\left(s_k^T g_{k+1}\right)}{s_k^T y_K} &\le \frac{1}{2} \frac{\frac{1}{2}\left(s_k^T y\right)^2 g_{k+1}^T g_{k+1} + 2\left(s_k^T g_{k+1}\right)^2 y_k^T y_k}{\left(s_u^T y_k\right)^2} \\ &= \frac{1}{4}\|g_{k+1}\|^2 + \frac{\left(s_k^T g_{k+1}\right)^2 y_k^T y_k}{\left(s_k^T y_k\right)^2} \\ &\le \frac{1}{4}\|g_{k+1}\|^2 + \frac{L\left(s_k^T g_{k+1}\right)^2}{s_k^T y_k} \end{aligned} \tag{38}$$

Use (38) in (37) to get:

$$\begin{aligned} d_{k+1}^T g_{k+1} &\le -\|g_{k+1}\|^2 + \frac{1}{4}\|g_{k+1}\|^2 + L\frac{\left(s_k^T g_{k+1}\right)^2}{s_k^T y_k} \\ &= -\frac{3}{4}\|g_{k+1}\|^2 + L\frac{\left(s_k^T g_{k+1}\right)^2}{s_k^T y_k} \end{aligned} \tag{39}$$

Hence the search directions are descent if (36) satisfied and the proof of the theorem (3) is completed.

RESULTS

Here we reported some numerical results obtained with the implementation of the new v1 and v2 algorithms on a set of unconstrained optimization test problems taken from[2,3]. We have selected (15) large scale unconstrained optimization problems in extended or generalized form; for each test function we have considered numerical experiments with the number of variables n = 100, 1000, 10000.

These two new versions are compared with two well-know CG-algorithms; the first is the Hestenes and Stiefel (HS) algorithm which is one of the best and well-known CG-algorithms[5] in practice and always generates conjugate directions independent of line search and objective functions. The second is the original Fletcher and Reeves (FR) algorithm. All these algorithms are implemented with the standard Wolfe line search conditions (10a) and (10b) with $c_1 = 0.0001$ and $c_2 = 0.9$ where initial step-size $\alpha_k = 1/\|g_0\|$ and the initial guess for other iterations i.e., (k>0):

$$\alpha_k = \alpha_{k-1} * \sqrt{\|d_{k-1}\| / \|d_k\|} \tag{40}$$

In the all these cases, the stopping criteria is the $\|g_k\| \le 10^{-6}$. Problems numbers indicat for: 1 is the Extend trigonometric, 2 is the Extend Rosenbrok, 3 is the Penlty, 4 is the Perturbed Quadratic, 5 is the Rayadan 1, 6 is the Extended three exponential terms, 7 is the Generalized tridigonal 2, 8 is the Extended Powell, 9 is the Extended wood, 10 is the Quadratic QF1, 11 is the Quadratic QF2, 12 is the Extend tridigonal 2, 13 is the Almost pertumbed quadratic, 14 is the Tridiognal perturbed quadratic, 15 is the ENGAL1 (CUTE).

Because the main costs in the numerical optimization are the Function And Gradient Evaluations (FGEV) and also the number of Iterations (IT), hence our comparison are based on the function, gradient evolutions (which they are equal in these CG-algorithms by employing cubic fitting technique as a line search subprogram). Also in the comparison we considered the ability of the algorithms to solve particular test problems.

All codes are written in double precision FORTRAN (2000) with F77 default compiler settings. These codes are originally written by Andrei[1,2] and modified by the authors.

Table 1: Comparison of different CG-algorithms with different test functions and different dimensions

		HS algorithm		New V1 algorithm		New V2 algorithm		FR algorithm	
P. No.	n	IT	FGEV	IT	FGEV	IT	FGEV	IT	FGEV
1	100	19	35	21	38	18	33	19	35
	1000	39	67	36	67	31	56	38	65
	10000	34	59	39	72	34	59	32	60
2	100	34	72	34	72	32	70	47	93
	1000	35	77	35	81	34	74	78	131
	10000	35	83	35	82	35	82	54	106
3	100	9	25	11	29	11	29	10	27
	1000	329	10306	22	49	49	902	24	191
	10000	19	259	99	2791	14	42	92	2406
4	100	102	155	86	130	110	168	95	150
	1000	380	595	368	581	353	543	349	568
	10000	1092	1703	1064	1664	1203	1879	1417	2160
5	100	80	122	91	146	99	156	102	161
	1000	390	720	372	658	339	602	*	*
	10000	*	*	1442	2516	1402	2596	*	*
6	100	14	23	15	23	13	22	15	25
	1000	30	435	12	21	14	23	127	3531
	10000	147	4175	91	2350	84	2131	164	4620
7	100	42	62	37	29	36	58	37	67
	1000	67	102	61	100	63	98	73	115
	10000	57	96	64	100	64	102	180	300
8	100	75	141	98	181	89	169	180	313
	1000	75	143	121	227	156	288	*	*
	10000	81	153	160	294	123	228	*	*
9	100	32	60	34	66	32	60	71	110
	1000	28	54	32	62	28	54	47	84
	10000	31	61	36	69	43	80	47	86
10	100	100	152	97	149	95	142	108	174
	1000	353	542	333	518	385	598	313	520
	10000	1106	1731	1061	1670	1113	1759	1193	1742
11	100	119	180	105	166	104	165	130	196
	1000	396	619	357	561	362	572	364	593
	10000	1668	2468	1236	1967	1307	2089	1839	2905
12	100	38	61	44	70	39	63	40	65
	1000	40	64	41	66	39	63	34	68
	10000	284	8112	358	8415	198	5328	160	3964
13	100	104	157	85	120	92	144	98	157
	1000	365	570	311	489	317	493	314	519
	10000	1241	1970	1204	1892	1240	1932	1276	1981
14	100	26	47	101	164	97	153	106	166
	1000	82	1664	337	530	341	535	335	941
	10000	1186	1847	1181	1842	1261	1983	1187	1846
15	100	29	50	20	50	27	47	34	57
	1000	81	1940	67	1423	93	2192	142	3616
	10000	213	6245	134	3650	253	7601	203	5655

*: The algorithmfail to converge

Table 2: Comparison of different CG-algorithms with respect to the number of best (IT and FG)

	FR algorithm	HS algorithm	New v1 algorithm	New v2 algorithm	Equivalence relations
n	IT(FGEV)	IT(FGEV)	IT(FGEV)	IT(FGEV)	IT(FGEV)
100	-(-)	5(5)	3(2)	6(7)	1(1)
1000	3(-)	2(1)	6(7)	3(6)	1(1)
10000	3(-)	2(3)	6(7)	3(2)	1(3)

In Table 1 we have compared HS, FR, new v1 and new v2 algorithms for three values of n. The symbol * in Table 1 means that the algorithm is unable to solve the particular problem in less than the maximum number of iterations which it is 2000 in our comparisons.

From Table 1 we have observed that the new v2 algorithm is the better than the others for n = 100, in terms of the number of results against (IT and FGEV). Details of the best results of these compared algorithms are shown in Table 2. From this Table we have observed also that for n = 1000, the new v1 is also the best and for n = 10000, the new v1 is the best, overall 45 problem dimensions test operations.

CONCLUSION

The suggested algorithms like the original CG-algorithms are gave better numerical results in terms of IT and FGEV which are clearly well-defined from Table 2.

We have observed by the two new versions of the CG-algorithms which are suggested in this study that they are arrive at the limit point while the original HS and FR algorithms are failed.

REFERENCES

1. Andrei, N., 2008. 40 conjugate gradient algorithm for unconstrained optimization. A survey on their definition ICI. Technical Report No. 13/08. http://www.ici.ro/camo/neculai/p13a08.pdf
2. Andrei, N., 2008. An unconstrained optimization test functions collection. Adv. Model. Optimiz., 10: 147-161. http://www.ici.ro/camo/journal/vol10/v10a10.pdf
3. Bongartz, I., A.R. Conn, N.I.M. Gold and P.L. Toint, 1995. CUTE: Constrained and unconstrained testing environment. ACM. Trans. Math. Software, 21: 123-160. DOI: 10.1145/200979.201043
4. Dai, Y.H. and Y. Yuan, 1999. A nonlinear conjugate gradient algorithm with a strong global convergence property. SIAM. J. Optimiz., 10: 177-182. DOI: 10.1137/S1052623497318992
5. Dai, Y.H. and L.Z. Liao, 2001. New conjugacy conditions and related nonlinear conjugate gradient algorithms. Applied Math. Optimiz., 43: 87-101. DOI: 10.1007/s002450010019
6. Fletcher, R., 1987. Practical algorithms for Unconstrained Optimization. 2nd Edn., John Wiley and Sons, Chichester, ISBN: 0 471 91547 5, pp: 451.
7. Fletcher, R., 1993. An overview of unconstrained optimization. Numerical Analysis Report, NA/149. http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.47.3716&rep=rep1&type=pdf
8. Hagar, W.W. and H. Zhang, 2006. A survey of nonlinear conjugate gradient algorithms. Pacific J. Optimiz., 2: 35-58. http://www.math.lsu.edu/~hozhang/papers/cgsurvey.pdf
9. Han, J., 2001. Two fundamental convergence theorems for nonlinear conjugate gradient algorithms and their applications. Acta Math. Appli. Sinica 17: 38-46. DOI: 10.1007/BF02669682
10. Hiroshi, Y. and M. Takano, 2004. Global convergence properties of nonlinear conjugate gradients with modified secant condition. Comput. Optimiz. Appli., 28: 203-225. DOI: 10.1023/B:COAP.0000026885.81997.88
11. Jean, G. and N. Jorge, 1992. Global convergence properties of conjugate gradients algorithms for optimization. SIAM. J. Optimiz., 2: 21-42. http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.133.9688&rep=rep1&type=pdf
12. Shi, Z.J. and J. Guo, 2008. A new algorithm of nonlinear conjugate gradient method with strong convergence. Comput. Applied Math., 27: 93-106. http://www.scielo.br/pdf/cam/v27n1/a06v27n1.pdf
13. Wolfe, P., 1969. Convergence conditions for ascent algorithms. SIAM. Rev., 11: 226-235. http://www.jstor.org/pss/2028111
14. Zhen-Jun, S. and G. Jinhua, 2008. A new algorithm of nonlinear conjugate gradient algorithm with strong convergence. J. Comput. Applied Math., 27: 93-106. http://www.scielo.br/pdf/cam/v27n1/a06v27n1.pdf
15. Zhang, L., 2009. New versions of the Hestenes-Stiefel non-linear conjugate gradient algorithm based on the secant condition for optimization. J. Comput. Applied Math., 28: 111-133. http://www.scielo.br/pdf/cam/v28n1/a06v28n1.pdf
16. Zountendijk, G., 1970. Nonlinear Programming, Computational Algorithms. In: Integer and Nonlinear Programming, Abadie, J. (Ed.). North-Holland, Amsterdam, ISBN: 0444100008, pp: 37-86.