Journal of Computer Science 6 (5): 511-518, 2010
ISSN 1549-3636

New Scaled Sufficient Descent Conjugate Gradient Algorithm for Solving Unconstraint Optimization Problems

[1]Abbas Y. AL-Bayati and [2]Rafiq S. Muhammad
[1]Department of Mathematics, College of Computers Sciences and Mathematics, University of Mosul, Iraq
[2]Department of Mathematics, College of Education, University of Suleimani, Iraq

Abstract: Problem statement: The scaled hybrid Conjugate Gradient (CG) algorithm which usually used for solving non-linear functions was presented and was compared with two standard well-Known NAG routines, yielding a new fast comparable algorithm. **Approach:** We proposed, a new hybrid technique based on the combination of two well-known scaled (CG) formulas for the quadratic model in unconstrained optimization using exact line searches. A global convergence result for the new technique was proved, when the Wolfe line search conditions were used. **Results:** Computational results, for a set consisting of 1915 combinations of (unconstrained optimization test problems/dimensions) were implemented in this research making a comparison between the new proposed algorithm and the other two similar algorithms in this field. **Conclusion:** Our numerical results showed that this new scaled hybrid CG-algorithm substantially outperforms Andrei-sufficient descent condition (CGSD) algorithm and the well-known Andrei standard sufficient descent condition from (ACGA) algorithm.

Key words: Unconstrained optimization, hybrid conjugate gradient, scaled conjugate gradient, sufficient descent condition, conjugacy condition

INTRODUCTION

For solving the unconstrained optimization problem:

$$\min\left\{f(x): x \in R^n\right\} \quad (1)$$

where, $f: R^2 \to R$ is continuously differentiable function, bounded from below. Starting from an initial guess, a nonlinear CG-algorithm generates a sequence of points $\{x_k\}$, according to the following recurrence formula:

$$x_{k+1} = x_k + \alpha_k d_k \quad (2a)$$

where, α_k is the step-length, usually obtained by Wolfe line searches:

$$f(x_k + \alpha_k d_k) - f(x_k) \le \rho \alpha_k g_k^T d_k \quad (2b)$$

$$g_{k+1}^T d_k \ge \sigma g_k^T d_k \quad (2c)$$

with $0 < \rho < \frac{1}{2} \le \sigma < 1$ and the directions d_k are computed as:

$$d_0 = -g_0 \quad (3a)$$

$$d_{k+1} = -\theta_{k+1}^{CGSD} g_{k+1} + \beta_k^{CGSD} s_k \quad (3b)$$

Where:

$$y_k = g_{k+1} - g_k,\ s_k = x_{k+1} - x_k \quad (3c)$$

MATERIALS AND METHODS

Algorithms based on sufficient descent conditions: This type of algorithms present a modification of the standard computational CG scheme in order to satisfy both the sufficient descent and the conjugacy conditions in the frame of CG as in (4), with:

$$\theta_{k+1}^{CGSD} = \frac{g_{k+1}^T g_{k+1}}{y_k^T g_{k+1}} \quad (4)$$

$$\beta_k^{CGSD} = \frac{1}{y_k^T s_k}\left(g_{k+1} - \delta_k^{CGSD} \frac{\|g_{k+1}\|^2}{y_k^T s_k} s_k\right)^T g_{k+1} \quad (5)$$

$$\delta_k^{CGSD} = \frac{y_k^T g_{k+1}}{g_{k+1}^T g_{k+1}} \quad (6)$$

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

or:

$$\beta_k^{CGSD} = \frac{g_{k+1}^T g_{k+1}}{y_k^T s_k} - \frac{(y_k^T g_{k+1})(s_k^T g_{k+1})}{(y_k^T s_k)^2} \quad (7)$$

Equation 4-7 are represent an algorithm that belongs to the family of scaled CG-algorithms introduced by (Birgin and Martinez, 2001). Observing that if f is a quadratic function and α_k is selected to achieve the exact minimum of f in the direction d_k then $s_k^T g_{k+1} = 0$ and the formula (5) for β_k^{CGSD} reduced to the Dai and Yuan computational scheme (Andrei, 2008a):

$$\beta_k^{DY} = g_{k+1}^T g_{k+1} / y_k^T s_k \quad (8)$$

However, the parameter β_k^{CGSD} is considered for general non-linear functions and inexact line searches and it is selected in such a manner that the sufficient descent condition is satisfies at every iteration. Besides, the parameters θ_{k+1}^{CGSD} and δ_k are chosen in such manner that the conjugacy condition $y_k^T d_{k+1} = 0$ always holds, independently of the line searches used in the algorithm. Here below we list outlines of the Andrei algorithm.

CGSD algorithm (Andrei, 2007):

Step 1: Initialization. Select $x_0 \in R^n$ and the parameters $0 < \sigma_1 < \sigma_2 < 1$. Compute f ($x_0$) and g_0, consider $d_0 = -g_0$ and $\alpha_0 = \frac{1}{\|g_0\|}$, set k = 0.

Step 2: Test for convergence, if $\|g_0\| \le 10^{-6}$, then stop, else set k = k+1 and continue.

Step 3: Compute the line search parameter α_k which satisfy the Wolfe-conditions:

$$f(x_k + \alpha_k d_k) - f(x_k) \le \sigma_1 \alpha_k g_k^T d_k \quad (9)$$

$$\nabla f(x_k + \alpha_k d_k)^T d_k \ge \sigma_2 g_k^T d_k \quad (10)$$

update the variables $x_{k+1} = x_k + \alpha_k d_k$. Compute $f(x_{k+1})$:

$$g_{k+1},\ s_k = x_{k+1} - x_k,\ y_k = g_{k+1} - g_k \quad (11)$$

Step 4: Compute $d = -\theta_{k+1}^{CGSD} g_{k+1} + \beta_k^{CGSD} s_k$, where θ_{k+1}^{CGSD} and β_k^{CGSD} are defined as in (4) and (7) respectively.

Step 5: If $g_{k+1}^T d \le -10^{-3} \|d\|_2 \|g_{k+1}\|_2$, then define d_{k+1} = d, otherwise, set $d_{k+1} = -g_{k+1}$ and compute the initial guess $\alpha_k = \alpha_{k-1} \|d_{k-1}\|_2 / \|d_k\|_2$, then set k = k+1 and continue with Step 2.

ACGA algorithm (Andrei, 2009a): This algorithm also presents a modification of the Dai and Yuan computational scheme in order to satisfy both the sufficient descent and the conjugacy conditions in the frame of CG. The steps of this algorithm are same as in Andrei (CGSD), except Step (4) which will be defined as:

$$d_{k+1} = -g_{k+1} + \beta_k^{ACGA} s_k \quad (12a)$$

s.t.:

$$\beta_k = \frac{y_k^T g_{k+1}}{y_k^T s_k} - \frac{(y_k^T g_{k+1})(s_k^T g_{k+1})}{(y_k^T s_k)^2} \quad (12b)$$

Or:

$$\beta_k^{ACGA} = \frac{g_{k+1}^T g_k}{y_k^T s_k} - \delta_k^{ACGA} \frac{(g_{k+1}^T s_k)(g_{k+1}^T g_k)}{(y_k^T s_k)^2} \quad (12c)$$

Now we introduce a new proposed method based on modifying both scalars θ_{k+1} and β_k.

A new proposed algorithm (say, new hybrid): here, we are going to investigate another new sufficient descent algorithm based on the reformulation of the scalars θ_{k+1} and β_k. The new proposed scalars are depend on the general hybrid techniques of two or more than two parameters. These scalars are very useful in making the search directions generated by the new algorithm more sufficiently descent. The outlines of the new proposed algorithm are given by:

Step 1: Select $x_0 \in R^n$ and the parameters $0 < \sigma_1 < \sigma_2 < 1$. Compute f ($x_0$) and g_0, consider $d_0 = -g_0$ and $\alpha_0 = \frac{1}{\|g_0\|}$, set k = 0.

Step 2: Test for convergence, if $\|g_0\| \le 10^{-6}$, then stop, else set k = k+1 and continue.

Step 3: Compute the line search parameter α_k which satisfy the Wolfe-conditions defined by (9) and (10). Update $x_{k+1} = x_k + \alpha_k d_k$ Compute $f(x_{k+1})$, g_{k+1}.

Step 4: Compute $d = -\theta_{k+1}^{hybrid} g_{k+1} + \beta_k^{hybrid} s_k$, where θ_{k+1}^{hybrid} and β_k^{hybrid} are computed as:

$$\theta_{k+1}^{hybrid} = \max\left\{1.1\times 10^{-24}, \min\left\{1, \gamma_{k+1}, \theta_{k+1}^{CGSD}\right\}\right\} \quad (13a)$$

$$\gamma_{k+1} = \min\left\{\left\{\frac{d_k^T d_k(\alpha_k - \eta_{ki})^2}{2[f(x_{k+1}) - f(x_k) - (\alpha_k - \eta_k) g_k^T d_k]}\right\}_{i=1}^{10}\right\} \quad (13b)$$

s.t.:

$$\eta_{ki} = \frac{1}{d_k^T g_k}\left(f_{k+1} - f_k + \alpha_k g_k^T d_k + 10^{-i}\times\alpha_k^2\times\|g_k\|^2\right) \quad (13c)$$

$$\theta_{k+1}^{CGSD} = \frac{g_{k+1}^T g_{k+1}}{y_k^T g_{k+1}} \quad (13d)$$

$$\beta_k^{hybrid} = \max\left\{0, \min\left\{\beta_k^{CGSD}, \beta_k^{ACGA}\right\}\right\} \quad (14a)$$

$$\beta_k^{CGSD} = \frac{g_{k+1}^T g_{k+1}}{y^T s_k} - \frac{\left(y_k^T g_{k+1}\right)\left(s_k^T g_{k+1}\right)}{\left(y_k^T s_k\right)^2} \quad (14b)$$

$$\beta_k^{ACGA} = \frac{y_{k+1}^T g_{k+1}}{y_k^T s_k} - \frac{\left(y_k^T g_{k+1}\right)\left(s_k^T g_{k+1}\right)}{\left(y_k^T s_k\right)^2} \quad (14c)$$

where the details of β_k^{ACGA} are given in (Andrei, 2009a).

Step 5: If:

$$g_{k+1}^T d \le -10^{-3}\|d\|_2\|g_{k+1}\|_2 \text{ and } \left|g_{k+1}^T g_k\right| \le 0.2\|g_{k+1}\|_2^2 \quad (14d)$$

then define $d_{k+1} = d$, otherwise, set $d_{k+1} = -\theta_{k+1} g_{k+1}$ and compute the initial guess $\alpha_k = \alpha_{k-1}\|d_{k-1}\|_2 / \|d_k\|_2$, then set k = k+1 and continue with Step 2, where the details of (14d) are given in (Birgin and Martinez, 2001; Al-Bayati *et al*., 2009).

The algorithms (12) and (13) belongs to the family of hybrid CG-algorithms.

Rate of convergence of the new hybrid algorithm:

Theorem 1: If $y_k^T s_k \neq 0$ and:

$$d_{k+1} = -\theta_{k+1}^{CGSD} g_{k+1} + \beta_k^{CGSD} s_k \quad (14e)$$

$d_o = -g_o$ where β_k^{CGSD} is given by (5), then:

$$g_{k+1}^T d_{k+1} \le -\left(\theta_{k+1}^{CGSD} - \frac{1}{4\delta_k}\right)\|g_{k+1}\|^2 \quad (15)$$

Proof: Since d = -g_o, we have $g_o^T d_o = -\|g_o\|^2$, which satisfy (15). Multiplying (14e) by g_{k+1}^T, we have:

$$g_k^T d_{k+1} = -\theta_{k+1}^{CGSD}\|g_{k+1}\|^2 + \frac{\left(g_{k+1}^T g_{k+1}\right)\left(g_{k+1}^T s_k\right)}{y_k^T s_k} - \delta_k^{CGSD}\frac{\|g_{k+1}\|^2\left(s_k^T g_{k+1}\right)^2}{\left(y_k^T s_k\right)^2} \quad (16)$$

but:

$$\frac{\left[\left(y_k^T s_k\right) g_{k+1} / \sqrt{2\delta_k^{CGSD}}\right]^T\left[\sqrt{2\delta_k^{CGSD}}\left(y_k^T s_k\right) g_{k+1}\right]}{\left(y_k^T s_k\right)^2}$$
$$\le \frac{\frac{1}{2}\left[\frac{1}{2\delta_k^{CGSD}}\left(y_k^T s_k\right)^2\|g_{k+1}\|^2 + 2\delta_k^{CGSD}\left(g_{k+1}^T s_k\right)^2\|g_{k+1}\|^2\right]}{\left(y_k^T s_k\right)^2} = \quad (17)$$
$$\frac{1}{4\delta_k^{CGSD}}\|g_{k+1}\|^2 + \delta_k^{CGSD}\frac{\left(g_{k+1}^T s_k\right)^2\|g_{k+1}\|^2}{\left(y_k^T s_k\right)^2}$$

Using (17) in (16):

$$g_k^T d_{k+1} \le -\theta_{k+1}^{CGSD}\|g_{k+1}\|^2 + \frac{1}{4\delta_k^{CGSD}}\|g_{k+1}\|^2 + \delta_k^{CGSD}\frac{\left(g_{k+1}^T s_k\right)^2\|g_{k+1}\|^2}{\left(y_k^T s_k\right)^2} - \delta_k^{CGSD}\frac{\|g_{k+1}\|^2\left(s_k^T g_{k+1}\right)^2}{\left(y_k^T s_k\right)^2} \quad (18)$$

We get:

$$g_{k+1}^T d_{k+1} \le -\left(\theta_{k+1}^{CGSD} - \frac{1}{4\delta_k^{CGSD}}\right)\|g_{k+1}\|^2 \quad (19)$$

Hence, the direction given by (3) and (7) is a descent direction. If for all k, θ_{k+1}^{CGSD} is positive and given by $\theta_{k+1}^{CGSD} > \frac{1}{4\delta_k}$ and the line searches satisfy

Wolfe conditions, then the search directions given by (3) and (7) satisfy the sufficient descent condition since Andrei's algorithm bound by: $-\left(\theta_{k+1}^{CGSD}-\frac{1}{4\delta_k^{CGSD}}\right)\|g_{k+1}\|^2$.

Spectral θ_{k+1}^{CGSD} derivation: to determine the parameters θ_{k+1}^{CGSD} and δ_k^{CGSD} for CGSD method observe that:

$$d_{k+1}=-Q_{k+1}^{CGSD}g_{k+1} \tag{20}$$

Where:

$$d_{k+1}=-\theta_{k+1}^{CGSD}g_{k+1}+\left(\frac{g_{k+1}^Tg_{k+1}}{y_k^Ts_k}-\delta_k^{CGSD}\frac{\|g_{k+1}\|^2(s_k^Tg_{k+1})}{(y_k^Ts_k)^2}\right)\times s_k \tag{21}$$

$$d_{k+1}=-\left[\theta_{k+1}^{CGSD}I-\frac{s_kg_{k+1}^T}{y_k^Ts_k}+\delta_k^{CGSD}\frac{\|g_{k+1}\|^2}{(y_k^Ts_k)^2}(s_ks_k^T)\right].g_{k+1} \tag{22}$$

From (20) and (22) we get:

$$Q_{k+1}^{CGSD}=\theta_{k+1}^{CGSD}I-\frac{s_kg_{k+1}^T}{y_k^Ts_k}+\delta_k^{CGSD}\frac{\|g_{k+1}\|^2}{(y_k^Ts_k)^2}(s_ks_k^T) \tag{23}$$

Now, by summarization of Q_{k+1}^{CGSD} as:

$$\overline{Q}_{k+1}^{CGSD}=\theta_{k+1}^{CGSD}I-\frac{s_kg_{k+1}^T+g_{k+1}s_k^T}{y_k^Ts_k}+\delta_k^{CGSD}\frac{\|g_{k+1}\|^2}{(y_k^Ts_k)^2}(s_ks_k^T) \tag{24}$$

And considering the conjugacy condition:

$$y_k^Td_{k+1}=0 \tag{25}$$

$$y_k^T\overline{Q}_{k+1}^{CGSD}=0 \tag{26}$$

$$y_k^T\left[\theta_{k+1}^{CGSD}I-\frac{s_kg_{k+1}^T+g_{k+1}s_k^T}{y_k^Ts_k}+\delta_k^{CGSD}\frac{\|g_{k+1}\|^2}{(y_k^Ts_k)^2}(s_ks_k^T)\right]=0 \tag{27}$$

But:

$$\delta_k^{CGSD}=\frac{1}{\theta_{k+1}^{CGSD}} \tag{28}$$

After doing some algebraic operations, we get:

$$\left(\theta_{k+1}^{CGSD}\right)^2-\left(\frac{\|g_{k+1}\|^2}{y_k^Tg_{k+1}}+\frac{g_{k+1}^Ts_k}{y_k^Ts_k}\right)\theta_{k+1}^{CGSD}+\frac{(g_{k+1}^Ts_k)\|g_{k+1}\|^2}{(y_{k+1}^Tg_{k+1})(y_k^Ts_k)}=0 \tag{29a}$$

$$\left(\theta_{k+1}^{CGSD}-\frac{\|g_{k+1}\|^2}{y_k^Tg_{k+1}}\right)\left(\theta_{k+1}^{CGSD}-\frac{g_{k+1}^Ts_k}{y_k^Ts_k}\right)=0 \tag{29b}$$

Since $\|g_{k+1}\|^2$ in the numerator of CG operators has a strong global convergence (Al-Bayati *et al.*, 2009), hence from the first bracket of the Eq. 29b:

$$\theta_{k+1}^{CGSD}=\frac{\|g_{k+1}\|^2}{y_{k+1}^Tg_{k+1}} \tag{29c}$$

$$\delta_k^{CGSD}=\frac{y_{k+1}^Tg_k}{g_{k+1}^Tg_{k+1}}=\frac{1}{\theta_{k+1}^{CGSD}} \tag{30}$$

From (29c) we have observed that:

$$\theta_{k+1}^{CGSD}-\frac{1}{4\delta_k^{CGSD}}=\frac{3}{4}\theta_{k+1}^{CGSD} \tag{31a}$$

Therefore, for all k , $\theta_{k+1}^{CGSD}\geq 0$, i.e. if $g_{k+1}^Ty_k>0$, then for all k the search direction d_{k+1} given by (3) and (7) with (33), given later, satisfy the sufficient descent condition.

Anticipative θ_{k+1} derivation: Recently (Andrei, 2004) using the information in two successive points of the iterative process, proposed another approximation scalar to the Hessian matrix of function f , to obtain a new algorithm which was favorably compared with the Barzilai and Browein's method. This is only a half step of the spectral procedure. Indeed, at the point $x_{k+1}=x_k+\alpha_kd_k$, we can write:

$$f(x_{k+1})=f(x_k)+\alpha_kg_k^Td_k+\frac{1}{2}\alpha_k^2d_k^T\nabla^2f(z)d_k \tag{31b}$$

where, z is on the line segment connecting x_k and x_{k+1}. Having in view the local character of the searching procedure and that the distance between x_k and x_{k+1} is small enough, we can choose z = x_{k+1} and consider $\gamma_{k+1}\in R$ as a scalar approximation of $\nabla^2f(x_{k+1})$. This is an anticipative viewpoint, in which a scalar approximation of the Hessian at point x_{k+1} is computed using only the local information from two successive points: x_k and x_{k+1}, therefore we can write:

$$\gamma_{k+1}=\frac{2}{d_k^Td_k(\alpha_k)^2}[f(x_{k+1})-f(x_k)-\alpha_kg_k^Td_k] \tag{31c}$$

This formula can also be found in Dai an Yuan (Andrei, 2008b). Observing that $\gamma_{k+1}>0$ for convex functions (Andrei, 2007); if $f(x_{k+1})-f(x_k)-\alpha_kg_k^Td_k<0$, then the reduction $f(x_{k+1}) - f(x_k)$ in function values is smaller than $\alpha_k g_k^T d_k$. In this cases, the idea is to reduce the step size α_k as $\alpha_k - \eta_k$, maintaining the other quantities at their values in such away so that γ_{k+1} is positive. To get a value for η_k, let as select a real $\mu > 0$, "small enough" but comparable with the value of the function and have:

$$\eta_k = \frac{1}{g_k^T d_k}(f(x_k) - f(x_{k+1}) + \alpha_k g_k^T d_k + \mu) \tag{31d}$$

for which a new value of γ_{k+1} can be computed as:

$$\gamma_{k+1} = \frac{2}{d_k^T d_k (\alpha_k - \eta_k)^2}[f(x_{k+1}) - f(x_k) - (\alpha_k - \eta_k) g_k^T d_k] \tag{31e}$$

with these, the value for parameter θ_{k+1} is selected as:

$$\theta_{k+1} = \frac{1}{\gamma_{k+1}} \tag{31f}$$

where, γ_{k+1} is given by either (31c) or (31e).

Proposition: Assume that f(x) is continuously differentiable and $\nabla f(x)$ is Lipschitz continuous, with a positive constant L. then at point x_{k+1}:

$$\gamma_{k+1} \le 2L \tag{32}$$

Proof: From (31c) we have:

$$\gamma_{k+1} = \frac{2[f(x_k) + \alpha_k \nabla f(\zeta_k)^T d_k - f(x_k) - \alpha_k \nabla f(x_k)^T d_k]}{\|d_k\|^2 \alpha_k^2}$$

where, ζ_k is on the line segment connecting x_k and x_{k+1}. Therefore:

$$\gamma_{k+1} = \frac{2[\nabla f(\zeta_k) - \nabla f(x_k)]^T d_k}{\|d_k\|^2 \alpha_k}$$

Using the inequality of Cauchy and the Lipschitz continuity it follows that:

$$\gamma_{k+1} \le \frac{2\|\nabla f(\zeta_k) - \nabla f(x_k)\|}{\|d_k\| \alpha_k} \le \frac{2L\|\zeta_k - x_k\|}{\|d_k\| \alpha_k} \le \frac{2L\|x_{k+1} - x_k\|}{\|d_k\| \alpha_k} = 2L$$

Therefore, from (31f) we get a lower bound for $\theta_{k+1} \ge \frac{1}{2L}$, i.e., it is bounded away from zero.

Theorem 2: If $y_k^T s_k \ne 0$ and $d_{k+1} = -g_{k+1} + \beta_k^{ACGA} s_k$, ($d_0 = -g_0$), where β_k^{ACGA} is given by Eq. 12c, then:

$$g_{k+1}^T d_{k+1} \le -(1 - \frac{1}{4\delta_k^{ACGA}})\|g_{k+1}\|^2 \tag{33}$$

Proof: Since $d_0 = -g_0$, we have $g_0^T d_0 = -\|g_0\|^2$, which satisfies Eq. 33. Multiplying Eq. 12a by g_{k+1}^T, we have:

$$g_{k+1}^T d_{k+1} = -\|g_{k+1}\|^2 + \frac{(g_{k+1}^T g_{k+1})(g_{k+1}^T s_k)}{y_k^T s_k} - \delta_k^{ACGA} \frac{\|g_{k+1}\|^2 (g_{k+1}^T s_k)}{(y_k^T s_k)^2} \tag{34}$$

But:

$$\frac{(g_{k+1}^T g_{k+1})(g_{k+1}^T s_k)}{y_k^T s_k} = \frac{[(y_k^T s_k) g_{k+1} / \sqrt{2\delta_k^{ACGA}}]^T [\sqrt{2\delta_k^{ACGA}} (g_{k+1}^T s_k) g_{k+1}]}{(y_k^T s_k)^2} \tag{35}$$

$$\le \frac{\frac{1}{2}\left[\frac{1}{2\delta_k^{ACGA}}(y_k^T s_k)^2 \|g_{k+1}\|^2 + 2\delta_k^{ACGA}(g_{k+1}^T s_k)^2 \|g_{k+1}\|^2\right]}{(y_k^T s_k)^2} \tag{36}$$

$$= \frac{1}{4\delta_k^{ACGA}}\|g_{k+1}\|^2 + \delta_k^{ACGA} \frac{(g_{k+1}^T s_k)^2 \|g_{k+1}\|^2}{(y_k^T s_k)^2} \tag{37}$$

Using Eq. 37 in 34 we get Eq. 33.

Hence, the direction given by (12) is a descent direction because $(1 - 1/4\delta_k^{ACGA}) > 0$ for all k.

How to compute the parameter δ_k^{ACGA}: To determine the parameters δ_k^{ACGA} for (ACGA)-method observe that:

$$d_{k+1} = -Q_{k+1}^{ACGA} g_{k+1} \tag{38}$$

where:

$$d_{k+1} = -g_{k+1} + \frac{g_{k+1}^T g_{k+1}}{y_k^T s_k} s_k - \delta_k^{ACGA} \frac{\|g_{k+1}\|^2 (s_k^T g_{k+1})}{(y_k^T s_k)^2} s_k \quad (39)$$

where the matrix Q_{k+1}^{ACGA} is:

$$Q_{k+1}^{ACGA} = I - \frac{s_k g_{k+1}^T}{y_k^T s_k} + \delta_k^{ACGA} \frac{\|g_{k+1}\|^2}{(y_k^T s_k)^2} (s_k s_k^T) \quad (40)$$

Now, by summarization of Q_{k+1}^{ACGA} to resemble the Quasi-Newton method, as:

$$\overline{Q}_{k+1}^{ACGA} = I - \frac{s_k g_{k+1}^T + g_{k+1} s_k^T}{y_k^T s_k} + \delta_k^{ACGA} \frac{\|g_{k+1}\|^2}{(y_k^T s_k)^2} (s_k s_k^T) \quad (41)$$

and considering the conjugacy condition:

$$y_k^T d_{k+1} = 0 \quad (42)$$

$$y_k^T \overline{Q}_{k+1}^{CGSD} g_{k+1} = 0 \quad (43)$$

$$y_k^T [I - \frac{s_k g_{k+1}^T + g_{k+1} s_k^T}{y_k^T s_k} + \delta_k^{ACGA} \frac{\|g_{k+1}\|^2}{(y_k^T s_k)^2} (s_k s_k^T)] g_{k+1} = 0 \quad (44)$$

After doing some algebraic operations, it follows that:

$$\delta_k^{ACGA} = \frac{y_k^T s_k}{g_{k+1}^T s_k} + \frac{g_{k+1}^T y_k}{\|g_{k+1}\|^2} - \frac{(g_{k+1}^T y_k)(y_k^T s_k)}{\|g_{k+1}\|^2 (g_{k+1}^T s_k)} \quad (45)$$

Therefore using (45) in (12c) we get (12b).

RESULTS

We present the computational performance of a Fortran implementation of the new hybrid algorithm on a set of 1915 unconstrained optimization test problems/dimensions. The Fortran implementation of the present algorithm is based on the Fortran 90 implementation of the scaled CG-method provided by (Birgin and Martinez, 2001). The comparisons of algorithms are given in the following context. Let $f_i^{ALG!}$ and f_i^{ALG2} be the optimal values found by ALG1 and ALG2, for problem i = 1.., 65, respectively. We say that, in a particular problem i, the performance of ALG1 was better than the performance of ALG2 if:

$$\left| f_i^{ALG1} - f_i^{ALG2} \right| < 10^{-3} \quad (46)$$

and the number of iterations, or the number of function-gradient evaluations, of ALG1was less than the number of iterations, or the number of function-gradient evaluations of ALG2, respectively (Andrei, 2009a; 2009b; 2008c). We compare the performance of our new hybrid algorithm against the CGSD-algorithm (Andrei, 2008a) and against the standard ACGA-algorithm (Andrei, 2009a) in three different tables. In Table 1 and 2, sixty-five test-functions are solved using three different algorithms; namely: (Andrei, 2007) (CGSD); (Andrei, 2009a), (ACGA) and the new proposed (New Hybrid) algorithms. Each test function is solved by using 10 different dimensions, n = 100, 200, …, 1000. Table 1 and 2 present the performances of these algorithms subject to the minimum number of iterations (# iter) and the minimum number of function-gradient evaluations (# fgev).

When comparing the new hybrid against CGSD in Table 1, subject to #iter, the new hybrid was better in 163 problems while CGSD was better in 134 problems; they are equal in 293 problems and fail in 60 problems out of 650 problems; now subject to #fgev, the New Hybrid was better in 168 cases while CGSD was better in 138 cases; they have equal results in 284 cases and fail in 60 cases.

In Table 2, according to #iter, the new hybrid algorithm was better in 274 cases while ACGA was better in 194 cases; they have equal results in 155 cases and fail in 54 cases. However, according to #fgev, New Hybrid was better in 273 cases while ACGA was better in 209 cases; they have equal results in 114 cases and fail in 54 cases.

Table 3 shows elaboration comparison of 6 arbitrary selected test functions with different dimensions out of the 65-test problems with the three different algorithms.

Table 1: Performance of the new hybrid versus CGSD; In 650 problem/dimension

	New hybrid	CGSD	Equality	Over	Total
#iter	163	134	293	60	650
#fgev	168	138	284	60	650

Table 2: Performance of the new hybrid versus ACGA; In 650 problem/dimension

	New hybrid	CGSD	Equality	Over	Total
#iter	247	194	155	54	650
#fgev	273	209	114	54	650

Table 3: Comparison of Different CG-algorithms with an arbitrary selection of 6 different test functions out of 65-test problems

		CGSD		New hybrid		ACGA	
Tf	n	iter	fgev	iter	fgev	iter	fgev
10	100	65	102	67	105	66	110
	200	91	135	97	143	91	136
	300	101	152	108	161	109	168
	400	145	223	143	216	125	182
	500	153	236	143	221	157	228
	600	144	214	175	246	171	249
	700	174	255	162	232	183	277
	800	179	264	214	315	185	270
	900	197	305	189	286	191	284
	1000	211	322	221	333	210	309
22	100	70	132	62	113	67	121
	200	112	212	62	114	117	221
	300	110	206	237	459	41	79
	400	139	255	125	235	95	170
	500	159	299	77	135	116	215
	600	65	121	55	99	119	225
	700	117	221	83	148	87	159
	800	86	159	111	200	165	302
	900	50	88	71	125	91	168
	1000	79	142	100	191	49	90
34	100	369	442	363	422	315	369
	200	568	645	512	573	513	576
	300	756	842	674	733	585	649
	400	868	963	819	884	790	858
	500	964	1039	881	947	781	831
	600	1059	1142	948	1023	907	970
	700	1101	1222	1000	1066	1020	1085
	800	1290	1421	1182	1256	1039	1102
	900	1522	1736	1215	1297	1174	1249
	1000	1378	1454	1316	1398	1196	1263
47	100	11	29	11	29	11	29
	200	15	37	15	37	15	37
	300	13	37	13	37	14	39
	400	15	40	15	40	15	40
	500	18	41	18	41	27	61
	600	15	40	15	40	24	61
	700	15	41	15	41	29	75
	800	16	43	16	43	23	57
	900	15	43	15	43	35	88
	1000	16	43	16	43	76	176
51	100	24	45	24	45	27	52
	200	27	56	27	56	24	47
	300	24	47	24	47	27	51
	400	24	48	24	48	25	51
	500	35	448	23	48	23	49
	600	23	49	23	47	39	551
	700	28	61	76	1608	25	47
	800	23	47	26	175	33	381
	900	21	41	21	41	31	281
	1000	25	49	31	282	25	52
65	100	32	52	35	56	39	65
	200	33	53	34	57	33	57
	300	37	56	36	57	36	58
	400	33	54	36	58	37	62
	500	34	58	37	58	35	58
	600	35	56	37	58	37	63
	700	33	59	33	57	35	59
	800	32	52	36	56	32	58
	900	34	55	36	56	35	62
	1000	33	57	32	55	36	62

Finally, we have selected (65) large-scale unconstrained optimization problems in (10) different dimensions and in generalized from the CUTE (Bongartz *et al*., 1995) library, along with other large-scale optimization problems.

DISCUSSION

In this study, we have introduced a new scaled hybrid (CG) algorithm which is based on two well-known (CG) formulas. The new algorithm is compared with two well-known libraries; namely CGSD and ACGA algorithms using (65) well-known non linear test functions with (10) different dimensions. Our numerical results indicate that the new technique has an improvements of about (5%) in both #iter and #fgev against the standard CGSD algorithm. While it saves about (6%) in both #iter and #fgev against the standard ACGA algorithm. The name of test functions are: given in (Bongartz *et al*., 1995).

1. Freudenstien and Roth function:
2. Extended Trigonometric Function
3. Extended Rosenbrock Function:
4. Extended White and Holst function:
5. Extended Beal function:
6. Extended penalty function:
7. Peturbed Quadratic function.
8. Raydan 2 Function
9. Diagonal 1, 2 and 3 Functions.
10. Diagonal 2 Function.
11. Diagonal 3 Function.
12. Hager Function.
13. Generalized Tridiagonal-1 Function.
14. Extended Tridiagonal-1 Function.
15. Extended Three Exponential Terms.
16. Generalized Tridiagonal-2 Function.
17. Diagonal4 Function.
18. Diagonal5 Function (Matrix Rom).
19. HIMMELBC (CUTE).
20. Generalized PSC1 function.
21. Extended PSC1 Function.
22. Extended Powell Function.
23. Extended Block Diagonal BD1 Function.
24. Extended Maratos function.
25. Extended Cliff CLIFF (CUTE).
26. Quadratic Diagonal Perturbed Function
27. Extended Wood Function:
28. Quadratic Function QF
29. Extended Quadratic Penalty QP1 Function
30. A Quadratic Function QF2
31. Extended EP1 Function
32. Extended Tridiagonal-2 Function

33. BDQRTIC Function:
34. TRIDIA Function:
35. NONDQUAR Function:
36. DQDRTIC Function:
37. EG2 function:
38. DIXMAANA (CUTE)
39. DIXMAANB (CUTE)
40. DIXMAANC (CUTE)
41. DIXMAANE (CUTE):
42. Partial Perturbed Quadratic
43. Broyden Tridiagonal Function:
44. Almost Perturbed Quadratic Function:
45. Tridiagonal Perturbed Quadratic
46. EDENSCH Function (CUTE)
47. Vardim Function (Cute):
48. STAIRCASE S1Function:
49. DIAGONAL 6
50. DIXON3DQ Function:
51. ENGVAL1 (CUTE) Function:
52. DENSCHNA (CUTE) Function:
53. DENSCHNB (CUTE) Function:
54. DENSCHNF (CUTE) Function:
55. SINQUAD (CUTE) Function:
56. BIGGSB1 Function(Cute):
57. Generalized quartic GQ1 function:
58. Diagonal 7 Function:
59. DIAGONAL8 Function :
60. Full Hessian Function:
61. SINCOS Function:
62. Generalized quartic GQ2 function
63. EXTROSNB(CUTE):
64. ARGLINB (CUTE)
65. HIMMELBG (CUTE)

CONCLUSION

In this research, a new fast scaled hybrid CG algorithm is introduced. The proposed algorithm improved the standard CGSD and ACGA algorithms by adaptively modifying the search direction. The new proposed algorithm is generic and easy to implement in all gradient based optimization process. The simulation results showed that it is robust and has a potential significantly enhance the computational efficiency of iterations and function-gradient evaluations.

REFERENCES

Al-Bayati, A.Y., A.J. Salim and K.K. Abbo, 2009. Two-version of conjugate gradient-algorithms based on conjugacy condition for unconstrained optimization. Am. J. Econ. Bus. Admin., 1: 97-104 http://www.scipub.org/fulltext/ajeba/ajeba1297-104.pdf

Andrei, N., 2004. A new gradient descent method for unconstrained optimization. ICI, Technical report, Bucharest1, Romania. http://www.ici.ro/camo//neculai/newstep.pdf

Andrei, N., 2007. Scaled memory less BFGS preconditioned conjugate gradient algorithm for unconstrained optimization. Methods Soft Ware, 22: 561-571. DOI: 10.1080/10556780600822260

Andrei, N., 2008a. A Dai-Yuan conjugate gradient algorithm with sufficient descent and conjugacy conditions for unconstrained optimization. Applied Math. Lett., 21: 165-171. DOI: 10.1016/J.AML.2007.05.002

Andrei, N., 2008b. A scaled nonlinear conjugate gradient algorithm for unconstrained optimization. ICI Tech. Rep., 57: 549-570. DOI: 10.1080/02331930601127909

Andrei, N., 2008c. 40 Conjugate gradient algorithms for unconstrained optimization. ICI, Technical Report No. 13/08. http://www.ici.ro/camo/neculai/p13a08.pdf

Andrei, N., 2009a. Another nonlinear conjugate gradient algorithm for unconstrained optimization. Methods Software, 24: 89-104. DOI: 10.1080/10556780802393326

Andrei, N., 2009b. Performance profiles of line-search algorithms for unconstrained optimization. ICI, Technical Report. http://www.ici.ro/camo/neculai/w07p27.pdf

Birgin, B. and J.M. Martinez, 2001. A spectral conjugate gradient method for unconstrained optimization. Applied Math. Optimiz., 43: 117-128. DOI: 10.1007/s00245-001-0003.

Bongartz, I., A.R. Conn, N.I. Gould and P. Toint, 1995. CUTE: constrained and unconstrained testing environments. ACM Trans. Math. Software, 21: 123-160. DOI: 10.1145/200979.201043.0