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

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

\[ \min \{ f(x) : x \in \mathbb{R}^n \} \tag{1} \]

where, \( f : \mathbb{R}^2 \rightarrow \mathbb{R} \) 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:

\[ x_{k+1} = x_k + \alpha_k d_k \tag{2a} \]

where, \( \alpha_k \) is the step-length, usually obtained by Wolfe line searches:

\[ f(x_k + \alpha_k d_k) - f(x_k) \leq \rho \alpha_k g_k^T d_k \tag{2b} \]

\[ g_k^T d_k \geq \sigma \| g_k \| d_k \tag{2c} \]

with \( 0 < \rho \leq \frac{1}{2} \leq \sigma < 1 \) and the directions \( d_k \) are computed as:

\[ d_k = -g_k \tag{3a} \]

\[ d_{k+1} = -\theta_{k+1} g_{k+1} + \beta_{k+1} s_k \tag{3b} \]

Where:

\[ y_k = g_{k+1} - g_k, \quad s_k = x_{k+1} - x_k \tag{3c} \]

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:

\[ \theta_{k+1} = \frac{g_{k+1}^T y_k}{y_k^T y_k} \tag{4} \]

\[ \rho_{k+1} = \frac{1}{y_k^T y_k} \left( g_{k+1}^T \frac{\| y_k^T y_k \|}{y_k^T y_k} - g_{k+1}^T y_k \right) y_k \tag{5} \]

\[ \delta_{k+1} = \frac{y_k^T g_{k+1}}{g_{k+1}^T g_{k+1}} \tag{6} \]
or:

\[ \beta_{k}^{\text{CGSD}} = \frac{g_{k+1}^{T}sb_{k+1}}{y_{k}^{T}s_{k}} - \frac{(y_{k}^{T}s_{k})^{2}}{(y_{k}^{T}s_{k})^{2}} \quad (7) \]

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 \( \alpha_{k} \) is selected to achieve the exact minimum of \( f \) in the direction \( d_{k} \) then \( s_{k}^Tg_{k+1} = 0 \) and the formula (5) for \( \beta_{k}^{\text{CGSD}} \) reduced to the Dai and Yuan computational scheme (Andrei, 2008a):

\[ \beta_{k}^{\text{DY}} = \frac{g_{k+1}^{T}sb_{k+1}/y_{k}^{T}s_{k}} \quad (8) \]

However, the parameter \( \beta_{k}^{\text{CGSD}} \) 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 \( \theta_{k+1}^{\text{CGSD}} \) and \( \delta_{k} \) 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 \( x_{0} \in \mathbb{R}^{n} \) and the parameters

\[ 0 < \sigma_{i} < \sigma_{2} < 1 \]. 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 \( \|g_{k}\| \leq 10^{-6} \), then stop, else set \( k = k+1 \) and continue.

Step 3: Compute the line search parameter \( \alpha_{k} \) which satisfy the Wolfe-conditions:

\[ f(x_{k} + \alpha_{k}d_{k}) - f(x_{k}) \leq \sigma_{1}\alpha_{k}g_{k}^{T}d_{k} \quad (9) \]

\[ \forall f(x_{k} + \alpha_{k}d_{k})^{T}d_{k} \geq \sigma_{2}g_{k}^{T}d_{k} \quad (10) \]

update the variables \( x_{k+1} = x_{k} + \alpha_{k}d_{k} \).

Compute \( f(x_{k+1}) \):

\[ g_{k+1} = x_{k+1} - x_{k} \quad \text{and} \quad g_{k+1} = \|g_{k+1}\| \]

Step 4: Compute \( d = -\theta_{k+1}^{\text{CGSD}}g_{k+1} + \beta_{k}^{\text{CGSD}}s_{k} \), where \( \theta_{k+1}^{\text{CGSD}} \) and \( \beta_{k}^{\text{CGSD}} \) are defined as in (4) and (7) respectively.

Step 5: If \( g_{k+1}^{T}d \leq 10^{-6}\|g_{k+1}\| \), then define \( d_{k+1} = d \), otherwise, set \( d_{k+1} = -g_{k+1} \) and compute the initial guess \( \alpha_{k+1} = \alpha_{k} \|g_{k+1}\|/\|d_{k+1}\| \), 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:

\[ d_{k+1} = -g_{k+1} + \beta_{k}^{\text{ACGA}}s_{k} \quad (12a) \]

s.t.:

\[ \beta_{k} = \frac{y_{k}^{T}s_{k}}{y_{k}^{T}s_{k}} - \frac{(y_{k}^{T}s_{k})^{2}}{(y_{k}^{T}s_{k})^{2}} \quad (12b) \]

Or:

\[ \beta_{k}^{\text{ACGA}} = g_{k+1}^{T}g_{k+1}/y_{k}^{T}s_{k} - \delta_{k}^{\text{ACGA}} \frac{(g_{k+1}^{T}s_{k})(g_{k+1}^{T}g_{k+1})}{(y_{k}^{T}s_{k})^{2}} \quad (12c) \]

Now we introduce a new proposed method based on modifying both scalars \( \theta_{k+1}^{\text{CGSD}} \) and \( \beta_{k}^{\text{CGSD}} \).

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 \( \theta_{k+1} \) and \( \beta_{k} \). 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 \( x_{0} \in \mathbb{R}^{n} \) and the parameters

\[ 0 < \sigma_{i} < \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 \( \|g_{k}\| \leq 10^{-6} \), then stop, else set \( k = k+1 \) and continue.

Step 3: Compute the line search parameter \( \alpha_{k} \) which satisfy the Wolfe-conditions:

\[ f(x_{k} + \alpha_{k}d_{k}) - f(x_{k}) \leq \sigma_{1}\alpha_{k}g_{k}^{T}d_{k} \quad (9) \]

\[ \forall f(x_{k} + \alpha_{k}d_{k})^{T}d_{k} \geq \sigma_{2}g_{k}^{T}d_{k} \quad (10) \]

update the variables \( x_{k+1} = x_{k} + \alpha_{k}d_{k} \).

Compute \( f(x_{k+1}) \):

\[ g_{k+1} = x_{k+1} - x_{k} \quad \text{and} \quad g_{k+1} = \|g_{k+1}\| \]

Step 4: Compute \( d = -\theta_{k+1}^{\text{CGSD}}g_{k+1} + \beta_{k}^{\text{CGSD}}s_{k} \), where \( \theta_{k+1}^{\text{CGSD}} \) and \( \beta_{k}^{\text{CGSD}} \) are defined as in (4) and (7) respectively.
Step 4: Compute \( d = -\theta_{k+1}^{\text{hybrid}} g_{k+1} + \beta_k^{\text{hybrid}} s_k \), where \( \theta_{k+1}^{\text{hybrid}} \) and \( \beta_k^{\text{hybrid}} \) are computed as:

\[
\theta_{k+1}^{\text{hybrid}} = \max \left\{ 1, 10^{-24}, \min \{ 1, \gamma_{k+1}, \theta_k^{\text{CGSD}} \} \right\}
\]

\[
\gamma_{k+1} = \min \left\{ \left( \frac{d_k^T (\alpha_k - \eta_k) x_k}{2 f(x_{k+1}) - f(x_k) - (\alpha_k - \eta_k) g_k^T d_k} \right)^{10} \right\}
\]

Step 5: If:

\[
g_{k+1}^T d \leq -10^{-5} \| g_{k+1} \|_2 \quad \text{and} \quad \| g_{k+1} \|_2 \leq 0.2 \| g_{k+1} \|_2
\]

then define \( d_{k+1} = d \), otherwise, set \( d_{k+1} = -\theta_k g_{k+1} \) and compute the initial guess \( \alpha_k = \| d_k \|_2 / \| g_k \|_2 \), 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 \( y^T s_k \neq 0 \) and:

\[
d_{k+1} = -\theta_k^{\text{CGSD}} g_k + \beta_k^{\text{CGSD}} s_k
\]

\[
\theta_k = g_k^T d_k - \frac{1}{4 \delta_k} \| g_k \|_2^2
\]

\[
\beta_k = \frac{g_k^T d_k}{y_k^T s_k}
\]

\[
\beta_k^{\text{ACGA}} = \frac{y_k^T s_k}{y_k^T s_k} - \frac{g_k^T d_k}{y_k^T s_k}
\]

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

Proof: Since \( d = g_k \), we have \( g_k^T d_k = -\| g_k \|_2^2 \), which satisfy (15). Multiplying (14e) by \( g_k^T d_k \), we have:

\[
g_k^T d_{k+1} = -\theta_k^{\text{CGSD}} \| g_k \|_2^2 + \frac{(g_k^T d_k)(g_k^T s_k)}{y_k^T s_k}
\]

\[
= -\frac{g_k^T \| g_k \|_2}{(y_k^T s_k)^2}
\]

but:

\[
\left[ \left( y_k^T s_k / \sqrt{2 \delta_k^{\text{CGSD}}} \right)^2 \frac{\sqrt{2 \delta_k^{\text{CGSD}}} (y_k^T s_k)}{\sqrt{2 \delta_k^{\text{CGSD}}} (y_k^T s_k)} \right]
\]

\[
\leq \frac{1}{2} \left[ \frac{1}{\sqrt{2 \delta_k^{\text{CGSD}}} (y_k^T s_k)} \left( y_k^T s_k \right)^2 + 2 \delta_k^{\text{CGSD}} \left( y_k^T s_k \right)^2 \| g_k \|_2 \right]
\]

\[
= \frac{1}{4 \delta_k^{\text{CGSD}}} \| g_k \|_2^2 + \delta_k^{\text{CGSD}} \left( y_k^T s_k \right)^2 \| g_k \|_2^2
\]

Using (17) in (16):

\[
g_k^T d_{k+1} \leq -\theta_k^{\text{CGSD}} \| g_k \|_2^2 + \frac{1}{4 \delta_k^{\text{CGSD}}} \| g_k \|_2^2 + \delta_k^{\text{CGSD}} \left( y_k^T s_k \right)^2 \| g_k \|_2^2
\]

\[
\delta_k^{\text{CGSD}} \left( g_k^T d_k \right)^2 \| g_k \|_2^2 - \theta_k^{\text{CGSD}} \| g_k \|_2^2
\]

We get:

\[
g_{k+1}^T d_{k+1} \leq -\theta_k^{\text{CGSD}} \left( y_k^T s_k \right)^2 \| g_k \|_2^2
\]

Hence, the direction given by (3) and (7) is a descent direction. If for all \( k \), \( \theta_k^{\text{CGSD}} \) is positive and given by \( \theta_k^{\text{CGSD}} > \frac{1}{4 \delta_k} \) 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: 

$$
\theta^\text{CGSD}_{k+1} - \frac{1}{4\delta^\text{CGSD}} \left\| g_{k+1} \right\|_2^2
$$

Spectral $\theta^\text{CGSD}_{k+1}$ derivation: to determine the parameters $\theta^\text{CGSD}_{k+1}$ and $\delta_k^\text{CGSD}$ for CGSD method observe that:

$$
d_{k+1} = -Q^\text{CGSD}_{k+1} g_{k+1}
$$

Where:

$$
d_{k+1} = -\theta^\text{CGSD}_{k+1} g_{k+1} + \frac{g^T_{k+1} g_{k+1}}{y^T_{k} s_{k}} \cdot \delta_k^\text{CGSD} \left( \frac{y^T_{k} s_{k}}{y^T_{k} s_{k}} \right) x_{k+1}
$$

(21)

From (20) and (22) we get:

$$
Q^\text{CGSD}_{k+1} = \theta^\text{CGSD}_{k+1} I - \frac{g^T_{k+1} g_{k+1}}{y^T_{k} s_{k}} \cdot \delta_k^\text{CGSD} \left( \frac{y^T_{k} s_{k}}{y^T_{k} s_{k}} \right) \left( s_{k} s^T_{k} \right)
$$

(23)

Now, by summarization of $Q^\text{CGSD}_{k+1}$ as:

$$
\overline{Q}^\text{CGSD}_{k+1} = \theta^\text{CGSD}_{k+1} I - \frac{g^T_{k+1} g_{k+1}}{y^T_{k} s_{k}} + \delta_k^\text{CGSD} \left( \frac{y^T_{k} s_{k}}{y^T_{k} s_{k}} \right) \left( s_{k} s^T_{k} \right)
$$

(24)

And considering the conjugacy condition:

$$
y^T_{k} d_{k+1} = 0
$$

(25)

$$
y^T_{k} \overline{Q}^\text{CGSD}_{k+1} y_{k} = 0
$$

(26)

$$
y^T_{k} \left[ \theta^\text{CGSD}_{k+1} I - \frac{g^T_{k+1} g_{k+1}}{y^T_{k} s_{k}} + \delta_k^\text{CGSD} \left( \frac{y^T_{k} s_{k}}{y^T_{k} s_{k}} \right) \left( s_{k} s^T_{k} \right) \right] y_{k} = 0
$$

(27)

But:

$$
\delta_k^\text{CGSD} = \frac{1}{\theta^\text{CGSD}_{k+1}}
$$

(28)

After doing some algebraic operations, we get:

$$
\left( \theta^\text{CGSD}_{k+1} \right)^2 - \frac{\left\| g_{k+1} \right\|_2^2}{y^T_{k} g_{k+1}} + \frac{\left\| g_{k+1} \right\|_2^2}{y^T_{k} s_{k}} \cdot \theta^\text{CGSD}_{k+1} + \frac{\left\| g_{k+1} \right\|_2^2}{y^T_{k} g_{k+1}} \left( \frac{g^T_{k+1} s_{k}}{y^T_{k} s_{k}} \right) = 0
$$

(29a)

Since $\left\| g_{k+1} \right\|_2^2$ 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:

$$
\theta^\text{CGSD}_{k+1} = \frac{\left\| g_{k+1} \right\|_2^2}{y^T_{k} g_{k+1}}
$$

(29b)

$$
\delta_k^\text{CGSD} = \frac{1}{\theta^\text{CGSD}_{k+1}}
$$

(29c)

$$
\delta_k^\text{CGSD} = \frac{1}{\theta^\text{CGSD}_{k+1}}
$$

(30)

From (29c) we have observed that:

$$
\theta^\text{CGSD}_{k+1} = \frac{1}{4\delta^\text{CGSD}} = \frac{3}{4} \theta^\text{CGSD}_{k+1}
$$

(31a)

Therefore, for all $k$, $\theta^\text{CGSD}_{k+1} \geq 0$, i.e. if $g^T_{k+1} g_{k+1} > 0$, 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 $\theta^\text{ex} _{k+1}$ 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 $x_{k+1} = x_k + \alpha_k d_k$, we can write:

$$
f(x_{k+1}) = f(x_k) + \alpha_k g^T_k d_k + \frac{1}{2} \alpha_k^2 \left\| d_k \right\|_2^2
$$

(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 $\left\| d_k \right\|_2^2$. 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:

$$
\gamma_{k+1} = \frac{2}{d^T_k d_k (\alpha_k)^2} [f(x_{k+1}) - f(x_k) - \alpha_k g^T_k d_k]
$$

(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 $f(x_{k+1}) - f(x_k) - \alpha_k g^T_k d_k < 0$.  

514
then the reduction $f(x_{k+1}) - f(x_k)$ in function values is smaller than $\alpha_k g^T_k d_k$. In this cases, the idea is to reduce the step size $\alpha_k$ as $\alpha_k - \eta_k$, maintaining the other quantities at their values in such away so that $\gamma_{k+1}$ is positive. To get a value for $\eta_k$, let as select a real $\mu > 0$, “small enough” but comparable with the value of the function and have:

\[
\eta_k = \frac{1}{g^T_k d_k} (f(x_k) - f(x_{k-1}) + \alpha_k g^T_k d_k + \mu) \tag{31d}
\]

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

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

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

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

where, $\gamma_{k+1}$ 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 $x_{k+1}$:

\[
\gamma_{k+1} \leq 2L \tag{32}
\]

**Proof:** From (31c) we have:

\[
\gamma_{k+1} = \frac{2[\nabla f(x_{k+1}) - \nabla f(x_k)]^T d_k - f(x_k) - \alpha_k \nabla f(x_k)^T d_k]}{\|d_k\| \alpha_k} \tag{33}
\]

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

\[
\gamma_{k+1} = \frac{2[\nabla f(\zeta_k) - \nabla f(x_k)]^T d_k}{\|d_k\| \alpha_k} \tag{34}
\]

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

\[
\gamma_{k+1} \leq \frac{2[\nabla f(\zeta_k) - \nabla f(x_k)]}{\|d_k\| \alpha_k} \leq 2L \\|\zeta_k - x_k\| \tag{35}
\]

\[
\leq 2L \frac{\|x_{k+1} - x_k\|}{\|d_k\| \alpha_k} = 2L \tag{36}
\]

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

**Theorem 2:** If $y^T_k s_k \neq 0$ and $d_{k+1} = -g_{k+1} + \beta_k AC^A g_{k+1}$, ($d_0 = -g_0$), where $\beta_k$ is given by Eq. 12c, then:

\[
g^T_{k+1} d_{k+1} \leq -(1 - \frac{1}{4R^A C^A}) \|g_{k+1}\|^2 \tag{33}
\]

**Proof:** Since $d_0 = -g_0$, we have $g^T_0 d_0 = -\|g_0\|^2$, which satisfies Eq. 33. Multiplying Eq. 12a by $g^T_{k+1}$, we have:

\[
g^T_{k+1} d_{k+1} = -\|g_{k+1}\|^2 + \frac{g^T_0 g^T_{k+1} g_{k+1}^T s_k}{y^T_0 s_k} - \delta_{AC^A}^k \|g_{k+1}\|^2 \tag{34}
\]

But:

\[
\left( \frac{g^T_0 g^T_{k+1} g_{k+1}^T s_k}{y^T_0 s_k} \right)^T = \frac{y^T_0 s_k}{\|y^T_0 s_k\|} \left( \frac{2 R^A C^A g^T_{k+1} g_{k+1}^T s_k}{\|y^T_0 s_k\|^2} \right) \tag{35}
\]

\[
\leq \frac{1}{2} \left[ \frac{1}{2 R^A C^A \|g_{k+1}\|^2} \right] \|y^T_0 s_k\|^2 + \delta_{AC^A}^k \left( \frac{g^T_0 g^T_{k+1} g_{k+1}^T s_k}{y^T_0 s_k} \right) \|g_{k+1}\|^2 \tag{36}
\]

\[
= \frac{1}{4R^A C^A \|g_{k+1}\|^2} \|y^T_0 s_k\|^2 + \delta_{AC^A}^k \left( \frac{g^T_0 g^T_{k+1} g_{k+1}^T s_k}{y^T_0 s_k} \right) \|g_{k+1}\|^2 \tag{37}
\]

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

Hence, the direction given by (12) is a descent direction because $\left( \frac{1}{4R^A C^A} \right) > 0$ for all k.

**How to compute the parameter $\delta_{AC^A}^k$:** To determine the parameters $\delta_{AC^A}^k$ for (ACGA)-method observe that:

\[
d_{k+1} = -Q_{k+1} A_{k+1} \tag{38}
\]

where:
\[ d_{k+1} = -g_{k+1} + g^T_{k+1}g_{k+1} S_k - \delta^{ACGA}_{k} \left( \frac{s^T_{k} S_{k+1}}{(y^T_{k} S_k)^2} \right) S_k \]  

(39)

where the matrix \( Q^{ACGA}_{k+1} \) is:

\[
Q^{ACGA}_{k+1} = 1 - \frac{s^T_{k} S_{k+1}}{y^T_{k} S_k} + \delta^{ACGA}_{k} \left( \frac{s^T_{k} S_{k+1}}{(y^T_{k} S_k)^2} \right) \]  

(40)

Now, by summarization of \( Q^{ACGA}_{k+1} \) to resemble the Quasi-Newton method, as:

\[
\bar{Q}^{ACGA}_{k+1} = 1 - \frac{s^T_{k} S_{k+1}}{y^T_{k} S_k} + \delta^{ACGA}_{k} \left( \frac{s^T_{k} S_{k+1}}{(y^T_{k} S_k)^2} \right) \]  

(41)

and considering the conjugacy condition:

\[ y^T_{k} d_{k+1} = 0 \]  

(42)

\[ y^T_{k} Q^{CGSD}_{k+1} g_{k+1} = 0 \]  

(43)

\[ y^T_{k} \left[ 1 - \frac{s^T_{k} S_{k+1}}{y^T_{k} S_k} + \delta^{ACGA}_{k} \left( \frac{s^T_{k} S_{k+1}}{(y^T_{k} S_k)^2} \right) \right] g_{k+1} = 0 \]  

(44)

After doing some algebraic operations, it follows that:

\[
\delta^{ACGA}_{k} = \frac{y^T_{k} S_k + g^T_{k+1} y_{k+1}}{S_k - \|S_k\|^2} \frac{g^T_{k+1} y_{k+1} S_k}{\|S_k\|^2} \]  

(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 \( f^{ALG1}_{i} \) and \( f^{ALG2}_{i} \) 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:

\[
|f^{ALG1}_{i} - f^{ALG2}_{i}| < 10^{-3} \]  

(46)

and the number of iterations, or the number of function-gradient evaluations, of ALG1 was 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.

| Key | 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

| Key | New Hybrid | CGSD | Equality | Over | Total |
|-----|------------|------|----------|------|-------|
| #iter | 247 | 194 | 155 | 54 | 650 |
| #fgev | 273 | 209 | 114 | 54 | 650 |
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. Perturbed 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

| Tf  | n  | iter | fgev | iter | fgev | iter | fgev | iter | fgev |
|-----|----|------|------|------|------|------|------|------|------|
| 10  | 65 | 102  | 105  | 66   | 110  |
| 200 | 91 | 135  | 143  | 91   | 136  |
| 300 | 101 | 152  | 161  | 109  | 168  |
| 400 | 145 | 223  | 216  | 125  | 182  |
| 500 | 133 | 236  | 221  | 157  | 228  |
| 600 | 144 | 255  | 246  | 171  | 249  |
| 700 | 174 | 262  | 232  | 183  | 277  |
| 800 | 179 | 241  | 214  | 185  | 270  |
| 900 | 197 | 286  | 261  | 191  | 284  |
| 1000| 211| 333  | 310  | 210  | 309  |
| 22  | 70 | 132  | 113  | 67   | 121  |
| 200 | 112 | 212  | 114  | 117  | 221  |
| 300 | 110 | 206  | 145  | 41   | 79   |
| 400 | 139 | 255  | 235  | 95   | 170  |
| 500 | 159 | 299  | 135  | 116  | 215  |
| 600 | 65  | 121  | 99   | 119  | 225  |
| 700 | 117 | 221  | 148  | 87   | 159  |
| 800 | 86  | 159  | 111  | 205  | 302  |
| 900 | 50  | 88   | 125  | 91   | 168  |
| 1000| 79  | 142  | 191  | 49   | 90   |
| 34  | 369| 422  | 315  | 369  |
| 200 | 568 | 645  | 512  | 513  | 576  |
| 300 | 756 | 842  | 733  | 585  | 649  |
| 400 | 868 | 963  | 819  | 784  | 858  |
| 500 | 964 | 1039 | 947  | 781  | 831  |
| 600 | 1059| 1142 | 948  | 1023 | 907  | 970  |
| 700 | 1101| 1222 | 1066 | 1020| 1085  |
| 800 | 1290| 1421 | 1182 | 1256| 1102  |
| 900 | 1522| 1736 | 1215| 1174| 1249  |
| 1000| 1378| 1454 | 1316| 1196| 1263  |
| 47  | 11  | 29   | 29   | 11   | 29   |
| 200 | 15  | 35   | 37   | 15   | 37   |
| 300 | 13  | 37   | 37   | 14   | 39   |
| 400 | 15  | 40   | 40   | 15   | 40   |
| 500 | 18  | 41   | 41   | 27   | 41   |
| 600 | 15  | 40   | 40   | 24   | 46   |
| 700 | 15  | 41   | 41   | 29   | 75   |
| 800 | 16  | 43   | 43   | 23   | 57   |
| 900 | 15  | 43   | 43   | 35   | 88   |
| 1000| 16  | 43   | 43   | 76   | 176  |
| 51  | 24  | 45   | 45   | 27   | 52   |
| 200 | 27  | 56   | 56   | 24   | 47   |
| 300 | 24  | 47   | 47   | 27   | 51   |
| 400 | 24  | 48   | 48   | 25   | 51   |
| 500 | 35  | 448  | 43   | 23   | 49   |
| 600 | 23  | 49   | 47   | 39   | 551  |
| 700 | 28  | 61   | 67   | 1608 | 25   |
| 800 | 23  | 47   | 47   | 33   | 381  |
| 900 | 21  | 41   | 41   | 31   | 281  |
| 1000| 25  | 49   | 49   | 282  | 52   |
| 65  | 32  | 55   | 56   | 39   | 65   |
| 200 | 33  | 53   | 57   | 33   | 57   |
| 300 | 37  | 56   | 57   | 36   | 58   |
| 400 | 33  | 54   | 58   | 37   | 62   |
| 500 | 34  | 58   | 58   | 35   | 58   |
| 600 | 35  | 56   | 57   | 37   | 63   |
| 700 | 33  | 59   | 57   | 35   | 59   |
| 800 | 32  | 52   | 56   | 32   | 58   |
| 900 | 34  | 55   | 56   | 35   | 62   |
| 1000| 33  | 57   | 52   | 36   | 62   |
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/ajeja/ajeja1297-104.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 Gould and P. Toint. 1995. CUTE: constrained and unconstrained testing environments. ACM Trans. Math. Software, 21: 123-160. DOI: 10.1145/200979.201043.0