Indonesian Journal of Electrical Engineering and Computer Science
Vol. 20, No. 2, November 2020, pp. 939~947
ISSN: 2502-4752, DOI: 10.11591/ijeecs.v20.i2.pp939-947

A new hybrid conjugate gradient algorithm for unconstrained optimization with inexact line search

Fanar N. Jardow, Ghada M. Al-Naemi
Department of Mathematics, College of Computer Sciences and Mathematics, University of Mosul, Iraq

Article Info

Article history:

Received Feb 9, 2020
Revised Apr 5, 2020
Accepted Apr 19, 2020

Keywords:

Global convergence
Hybrid conjugate gradient method
Sufficient descent
SWP
Unconstrained optimization

ABSTRACT

Many researchers are interested for developed and improved the conjugate gradient method for solving large scale unconstrained optimization problems. In this work a new parameter θ_n will be presented as a convex combination between RMIL and MMWU. The suggestion method always produces a descent search direction at each iteration. Under Strong Wolfe Powell (SWP) line search conditions, the global convergence of the proposed method is established. The preliminary numerical comparisons with some others CG methods have shown that this new method is efficient and robust in solving all given problems.

Corresponding Author:

Ghada M. Al-Naemi,
Department of Mathematics, College of Computer Sciences and Mathematics,
University of Mosul, Mosul, Iraq.
Email: gh66alnaemi@gmail.com, drghadaalnaemi@uomosul.edu.eq

1. INTRODUCTION

Conjugate gradient (CG) methods are among the most popular methods for solving optimization problem, especially for large-scale problems due to the simplicity and low storage of their iterative form [1]. The unconstrained optimization problem has the following general form:

$$min_{x\in R^n}\, z(x) \tag{1}$$

where $x \in R^n$ is a real vector with $n \geq 1$ component and $z: R^n \to R$ is smooth function and its gradient g is available. The nonlinear CG method that starts from an initial guess $x_0 \in R^n$ will be defined using the iterations of the sequence as in the following form:

$$x_{n+1} = x_n + \rho_n d_n, \qquad n = 0,1,2,3,\ldots \tag{2}$$

where x_n is the n-th iterative point and ρ_n is the positive step size resulting from performing a one dimensional search, known as the line searches [2]. The d_n is the direction of the search that is computed by

$$d_{n+1} = \begin{cases} -g_{n+1}, & n = 0 \\ -g_{n+1} + \beta_n d_n, & n \geq 1 \end{cases} \tag{3}$$

where g_n record by $\nabla z(x_n)$ is the gradient and the $\beta_n > 0$ is a scalar known as the CG-coefficient, the different choices for the parameter β_n correspond to different conjugate gradient method. The step ρ_n length is very important for the global convergence of CG methods. It can either be exact or inexact. In the case of an exact steplength, one seeks ρ_n along the direction d_n such that

$f(x_n + \rho_n d_n) = min_{\rho>0} f(x_n + \rho d_n)$

For inexact ρ_n a number of line search techniques can be used. For instance,the so-called SWP condition require that [3, 4].

$$z(x_n + \rho_n d_n) \leq z(x_n) + \varepsilon_1 \rho_n d_n \quad (4)$$

$$|g(x_n + \rho_n d_n)| \leq \varepsilon_2 |g_n^T d_n| \quad (5)$$

where $0 < \varepsilon_1 < \frac{1}{2} < \varepsilon_2 < 1$, when x_n is far from the solution an approximation of ρ_n is found as the descending characteristic must be satisfied and the direction should not be searched. Thus by SWP we inherit the advantages of exact line search with inexpensive and low computational cost [5].

Different CG methods correspond to different choices of the parameter β_n [6]. The most popular formulas for parameters Hestenes Stiefel method (HS) [7]. Fletcher-Reeves method (FR) [8]. Polak-Ribiere – Polyak method (PR) [9, 10]. conjugate – Descent method (CD) [11]. Liu – Storey method (LS) [12]. and Dai-Yuan method (DY) [13]. The parameters of these β_n are given as follows:

$$\beta_n^{HS} = \frac{g_{k+1}^T y_k}{y_k^T d_k}; \qquad \beta_n^{FR} = \frac{g_{k+1}^T g_{k+1}}{g_k^T g_k}; \qquad \beta_n^{PRP} = \frac{g_{k+1}^T y_k}{g_k^T g_k};$$
$$\beta_n^{CD} = \frac{g_{k+1}^T g_{k+1}}{y_k^T d_k}; \qquad \beta_n^{LS} = \frac{g_{k+1}^T y_k}{-g_k^T d_k}; \qquad \beta_n^{DY} = \frac{g_{k+1}^T g_{k+1}}{y_k^T d_k}.$$

For a strictly convex quadratic function $z(x)$, and the line search is exact, all these methods are identical, since the gradients are mutually orthogonal, so the parameters β_n in these methods are equal. When implementated to general nonlinear function with inexact line searches, yet, the behavior of these methods is seeming different [14]. One of an important group of CG methods is the hybrid conjugate gradient algorithms, the hybrid computational schemes HCG work better than the classical CG methods because the HCG take the advantages of the two parameters β_n [15].

Many researchers devoted to the hybrid or mixed conjugate gradient methods which have better computational performances and strong convergence properties. Andrei [16] proposed the following hybrid method: $\beta_n^c = (1-\theta_n)\beta_n^{HS} + \theta_k \beta_n^{DY}$; Djordjevic' [17], proposed the following HCG method $\beta_n^{hyb} = (1-\theta_n)\beta_n^{LS} + \theta_n \beta_n^{FR}$; Xiuyun, *et al* [18], proposed the following HCG method $\beta_n^c = (1-\theta_n)\beta_n^{HS} + \theta_n \beta_n^{CD}$; Livieris, *et al* [19], proposed the following HCG method $\beta_n^{HCG} = \lambda_n \beta_n^{DY} + (1-\lambda_n)\beta_n^{HS}$; Al-Namat *et al* [20]. proposed the following HCG method $\beta_n^{FG} = (1-\theta_n)\beta_n^{MMWU} + \theta_n \beta_n^{RMAR}$.

In this work we focus on hybrid conjugate gradient methods as a convex combination of RMIL and MMWU [21, 22]. CG methods for solving unconstrained optimization method with suitable conditions. The corresponding conjugate gradient (CG) parameters are:

$$\beta_n^{RMIL} = \frac{g_{n+1}^T y_n}{\|d_n\|^2} \quad (6)$$

and

$$\beta_n^{MMWU} = \frac{\|g_{n+1}\|^2}{\|d_n\|^2} \quad (7)$$

The proposed method defined by set the parameter β_n by:

$$\beta_n^{HA} = (1-\theta_n)\beta_n^{RMIL} + \theta_n \beta_n^{MMWU} \quad (8)$$

Choosing the appropriate value of the θ_n in the convex combination, the search direction d_n of our algorithm not only is the Newton direction [23], so satisfies the famous DL conjugate condition proposed by Dai and Liao [24]. Under the SWP conditions, we prove the global convergence of the proposed algorithm, the numerical results also show the feasibility and activity of our algorithm. This study is organized as follows, Section 2 we introduce the new proposed hybrid CG method (HHA), and we got the parameter θ_n using some approaches and give us specific algorithm. Section 3, we prove that it generates direction satisfying the sufficient descent condition under SWP condition. Section 4, The global convergence property of the proposed method is established. in Section 5, Some numerical results are reported.

2. A NEW HYBRID CONJUGATE GRADIENT METHOD

In this section, we will describe a new proposed HCG method, in order to get the sufficient descent direction, we will compute θ_n as follows: we combine β_n^{RMIL} and β_n^{MMWU} in (8). The direction d_{n+1} are generated by:

$$d_{n+1} = -\, g_{n+1} + \beta_n^{HHA} d_n \tag{9}$$

The iterates $x_1, x_2, x_3, \ldots$ of the proposed method are computed by means of the recurrence (2), where the step size ρ_n is definition according to the SWP conditions (4) and (5). The scale parameter θ_n in (8) satisfying $0 \le \theta_n \le 1$, which will be determined a specific way to be described later. If $\theta_n \le 0$, then $\beta_n^{HHA} = \beta_n^{RMIL}$, and if $\theta_n \ge 1$, then $\beta_n^{HHA} = \beta_n^{MMWU}$. On the other hand, if $0 < \theta_n < 1$, then β_n^{HHA} is a convex combination of β_n^{RMIL} and β_n^{MMWU}. From (8) and (9) it is clear that:

$$d_{n+1} = \begin{cases} -g_{n+1}, & n = 1 \\ -g_{n+1} + (1-\theta_n)\frac{g_{n+1}^T y_n}{\|d_n\|^2} d_k + \theta_n \frac{\|g_{n+1}\|^2}{\|d_n\|^2} d_n, & n > 1 \end{cases} \tag{10}$$

Our motivation to select the parameter θ_n in such a manner that the defection d_{n+1} given (10) is equal to the Newton direction $d_{n+1}^N = -\nabla^2 f(x_{n+1})^{-1} g_{n+1}$. Therefore

$$-\nabla^2 f(x_{n+1})^{-1}\, g_{n+1} = -g_{n+1} + (1-\theta_n)\frac{g_{n+1}^T y_n}{\|d_n\|^2} d_n + \theta_n \frac{\|g_{n+1}\|^2}{\|d_n\|^2}\, d_n$$

$$-\nabla^2 f(x_{n+1})^{-1} g_{n+1} = -g_{n+1} + \frac{g_{n+1}^T y_n}{\|d_n\|^2} d_n - \theta_n \left(\frac{g_{n+1}^T y_n - \|g_{n+1}\|^2}{\|d_n\|^2}\right) d_n \tag{11}$$

Therefore, in order to have an algorithm for solving large scale problems we assume that pair (s_n, y_n) satisfies the secant equation. $y_n = \nabla^2 f(x_{n+1}) s_n$ so,

$$s_n^T \nabla^2 f(x_{n+1}) = y_n^T \tag{12}$$

Multiplying the as shown in (11) by $s_n^T\, \nabla^2 f(x_{n+1})$ from the left and denoting $\theta_n^{HA} = \theta_n$,we get

$$-s_n^T g_{n+1} = -s_n^T \nabla^2 \mathrm{f}(x_{n+1}) g_{n+1} + \frac{g_{n+1}^T y_n}{\|d_n\|^2} s_n^T \nabla^2 f(x_{n+1}) d_n - \theta_n^{HA}\left(\frac{g_{n+1}^T y_n - \|g_{n+1}\|^2}{\|d_n\|^2}\right) s_n^T \nabla^2 f(x_{n+1}) d_n$$

$$-s_n^T g_{n+1} = -y_n^T g_{n+1} + \frac{g_{n+1}^T y_n}{\|d_n\|^2} y_n^T d_n - \theta_n^{HA}\left(\frac{g_{n+1}^T y_n - \|g_{n+1}\|^2}{\|d_n\|^2}\right) y_n^T d_n$$

After some algebra, we get

$$\theta_n^{HA} = \frac{(s_n^T g_{n+1} - y_n^T g_{n+1})\|d_n\|^2 + (g_{n+1}^T y_n)(y_n^T d_n)}{(g_{n+1}^T g_n)(y_n^T d_n)} \tag{13}$$

We will specify a complete (HHA) which posses some nice properties of CG and Newton method.

ALGORITHM HHA

Step 1 : choose $\boldsymbol{x_0} \in \boldsymbol{R^n}$, $\in > \boldsymbol{0}$, Calculate $\boldsymbol{f(x_0)}$ and $\boldsymbol{g_0} = -\boldsymbol{\nabla f(x_0)}$, set $\boldsymbol{d_0} = -\boldsymbol{g_0}$, when $\boldsymbol{n = 0}$.
Step 2 : The stopping criteria, i.e. if $\|\boldsymbol{g_n}\| \le \in$, then stop.
Step 3 : Calculate $\boldsymbol{\rho_n}$ by SWP conditions in (3) & (4) .
Step 4 : Calculate $\boldsymbol{x_{n+1}} = \boldsymbol{x_n} + \boldsymbol{\rho_n d_n}$, and $\boldsymbol{g_{n+1}} = \boldsymbol{g(x_{n+1})}$.
Calculate $\boldsymbol{s_n} = \boldsymbol{x_{n+1}} - \boldsymbol{x_n}$ and $\boldsymbol{y_n} = \boldsymbol{g_{n+}} - \boldsymbol{g_n}$
Step 5 : If $\boldsymbol{\theta_n} \ge \boldsymbol{1}$ then put $\boldsymbol{\theta_n} = \boldsymbol{1}$. If $\boldsymbol{\theta_n} \le \boldsymbol{0}$, then put $\boldsymbol{\theta_n} = \boldsymbol{0}$, otherwise calculate $\boldsymbol{\theta_n}$ as (13).
Step 6 : Caclculate $\boldsymbol{\beta_n^{HA}}$ by (8).
Step 7 : Generate $\boldsymbol{d} = -\boldsymbol{g_{n+}} + \boldsymbol{\beta_n^{HA} d_n}$
Step 8 : If the restart criteria of Powell $|\boldsymbol{g_{n+1}^T g_n}| \ge \boldsymbol{0.2}\|\boldsymbol{g_{n+1}}\|^2$ is satisfied, then set $\boldsymbol{d_n} = -\boldsymbol{g_{n+1}}$, otherwise put $\boldsymbol{d_{n+1}} = \boldsymbol{d}$
Step 9 : put $n = n + 1$, and go to step 2.

3. THE SUFFICIENT DESCENT CONDITION

In this section, we use to the following theorem to clear up that the search direction d_n obtained by HHA satisfies the sufficient descent condition which plays of role in analyzing the global convergence. For further considerations we need the assumptions below:

3.1. Assumption

The level sets $Q = \{x: f(x) \le f(x_1)\}$ at x_1 is bounded where x_1 is starting point, namely, that there exists $M > 0$, such that $\|x\| \le M, \forall x \in Q$ [25].

3.2. Assumption

In a neighborhood N of Q, the function z is continuously differentiable and its gradient is Lipschitz continuous, i.e, there exists a constant $L > 0$, such that

$$\|\nabla z(x) - \nabla z(y)\| \le L\|x - y\|, \forall x, y \in N.$$

Under assumptions (3.1) and (3.2) , there exists positive constant ($u, \bar{u}, e$ & $\bar{e}$), such that:

$$\bar{u} \le \|g_{n+1}\| \le u, and\ \bar{e} \le \|g_n\| \le e \ \ \forall x \in Q \text{ [25].}$$

3.3. Theorem

Let generated the sequences $\{g_n\}$ and $\{d_n\}$ by a HHA method.then d_n is the search direction satisfies the sufficient descent condition:

$$g_{n+1}^T d_{n+1} \le -\tau\|g_{n+1}\|^2, \forall\, \tau \ge 0 \tag{14}$$

with $\tau = [\tau_3\tau_2 + (1 - \tau_3)\tau_1]$

3.4. Proof.

We show that search direction d_n shall satisfies the sufficient descent condition holds for $n = 0$, the proof is a trivial one, i.e. $d_0 = g_0$ and so $g_0^T d_0 = -\|g_0\|^2$. Now we have

$$d_{n+1} = -g_{n+1} + \beta_n^{HA} d_n,$$

$$d_{n+1} = -g_{n+1} + [(1 - \theta_n)\beta_n^{RMIL} + \theta_n\beta_n^{MMWU}]d_n\ .$$

We can rewrite the direction by the followin below:

$$d_{n+1} = -(\theta_n g_{n+1} + (1 - \theta_n)g_{n+1}) + ((1 - \theta_n)\beta_n^{RMIL} + \theta_n\beta_n^{MMWU})\mathrm{d_n}.$$

The above equation can be written after arrange the terms as:

$$d_{n+1} = \theta_n(-g_{n+1} + \beta_n^{MMWU} d_n) + (1 - \theta_n)(-g_{n+1} + \beta_n^{RMIL} d_n),$$

produces after some arrangement

$$d_{n+1} = \theta_n d_{n+1}^{MMWU} + (1 - \theta_n)d_{n+1}^{RMIL}\ , \tag{15}$$

produces after multiplying the (15) from the left by g_{n+1}^T, we get

$$g_{n+1}^T d_{n+1} = \theta_n g_{n+1}^T d_{n+1}^{MMWU} + (1 - \theta_n)g_{n+1}^T d_n^{RMIL} \tag{16}$$

Firstly, if $\theta_n = 0$, then $d_{n+1} = d_{n+1}^{RMIL}$, in [21] they proved that the sufficient descent condition holds with exact line search. We are going to prove that the sufficient descent condition holds for RMIL when inexact line search is used

$$g_{n+1}^T d_{n+1}^{RMIL} \le -\tau_1\|g_{n+1}\|^2, \tag{17}$$

where $\tau_1 = -(1 - 0.8\rho_n L) > 0$, with $0 < L < \frac{1}{0.8\,\rho_n}$.

Now let $\theta_n = 1$ then $d_n = d_n^{MMWU}$, in [22] they proved that the sufficient descent condition holds with exact line search. In [20], they proved that the sufficient descent condition holds with exact line search.

$$g_{n+1}^T d_{n+1}^{MMWU} \le -\tau_2 \|g_{n+1}\|^2 \ , \tag{18}$$

where $\tau_2 = (1 - \rho_n L) > 0$, with $0 < L < \frac{1}{\rho_n}$. Now, we are going to prove the direction satisfy the sufficient descent condition when $0 < \theta_n < 1$, we have $g_{n+1}^T s_n \le y_n^T s_n \le L\|s_n\|^2$, and $y_n = (g_{n+1} - g_n)$, then (13) become

$$\theta_n^{HA} \le \frac{(L\|s_n\|^2 - (\|g_{n+1}\|^2 - g_{n+1}^T g_n))\|d_n\|^2}{(g_{n+1}^T g_n)(y_n^T d_n)} + \frac{\|g_{n+1}\|^2 - g_{n+1}^T g_n}{g_{n+1}^T g_n}, \tag{19}$$

we have $v(1-\varepsilon_1)\|g_n\|^2 \le y_n^T d_n$, $y_n = (g_{n+1} - g_n)$, $|g_{n+1}^T g_n| \ge 0.2\|g_{n+1}\|^2$, and we know that

$$s_n = \rho_n d_n \Rightarrow d_n = \frac{s_n}{\rho_n} \Rightarrow \|d_n\| \le \frac{\|s_n\|}{|\rho_n|} \le \frac{\|x_{n+1} - x_n\|}{|\rho_n|} \le \frac{\|x_{n+1}\| + \|x_n\|}{|\rho_n|} \le \frac{M}{|\rho_n|} = D,$$

put the above in (19) become

$$\theta_n^{HA} \le \frac{1}{0.2}\left[\frac{(LM^2 - 0.8u^2)D^2}{v(1-\varepsilon_1)\bar{u}^2\bar{e}^2} + 0.8\right] = \tau_3 \tag{20}$$

From (15), (17), (18), and (20) we get

$$\therefore g_{n+1}^T d_{n+1} \le -[\tau_3\tau_2 + (1-\tau_3)\tau_1]\|g_{i+1}\|^2$$

$$\therefore g_{n+1}^T d_{n+1} \le -\tau\|g_{n+1}\|^2, \text{ with } \tau = [\tau_3\tau_2 + (1-\tau_3)\tau_1].$$

So, it is proved that d_{n+1} satisfied the sufficient descent condition

4. CONVERGE ANALYSIS

Let Assumption (3.1) and (3.2) hold. In [26] it is proved that for any conjugate gradient method with SWP conditions, it holds:

4.1. Lemma

Let Assumption (3.1) and (3.2) holds. Consider the method (2) and (5) where the d_n. Is a descent direction and ρ_n is received from the SWP. If

$$\sum_{n\ge1} \frac{1}{\|d_n\|^2} = \infty.$$

then

$$lim_{n\to\infty} \, inf\|g_n\| = 0 \ .$$

4.2. Theorem

Suppose that assumption (3.1) and (3.2) holds. Consider the algorithm HHA were $0 \le \theta_n \le 1$, and ρ_n is obtained by the strong wolfe line search and d_{n+1} is the descent direction. Then

$$lim_{n\to\infty} \, inf\|g_n\| = 0$$

4.3. Proof.

Because the descent condition holds, we have $d_{n+1} \ne 0$. So using lemma 4.1, it is sufficient to prove that $\|d_{n+1}\|$ is bounded above. From (10).

$$\|d_{n+1}\| = \|-g_{n+1} + [(1-\theta_n)\beta_n^{RMIL} + \theta_n\beta_n^{MMWU}]d_n\|$$

$$\le \|g_{n+1}\| + [|1-\theta_n|.|\beta_n^{RMIL}| + |\theta_n||\beta_n^{MMWU}|].\|d_n\|$$

They proved that in [21] and [22], that

$$|\beta_n^{RMIL}| \le \frac{g_{n+1}^T y_n}{\|d_n\|^2} \le \frac{\|g_{n+1}\|\|y_n\|}{\bar{B}^2} \le \frac{u\,A}{\bar{B}^2} = G_1 \ ,$$

And

$$|\beta_n^{MMWU}| = \frac{\|g_{n+1}\|^2}{\|d_n\|^2} \le \frac{u^2}{\bar{B}^2} = G_2 \ .$$

Now, we have

$$|\theta_n| = \left|\frac{(g_{n+1}^T s_n - y_n^T g_{n+1}).\|d_n\|^2 + (g_{n+1}^T y_n)(y_n^T d_n)}{(g_{n+1}^T g_n)(y_n^T d_n)}\right|.$$

Using SWC, we get $-(1-\varepsilon_1)\tau\|g_n\|^2 \le y_n^T d_n \le L\rho_n\|d_n\|^2$

$$|\theta_n| \le \left|\frac{(g_{n+1}^T s_n - y_n^T g_{n+1}).\|d_n\|^2 + (g_{n+1}^T y_n)L\rho_n\|d_n\|^2}{0.2\|g_{n+1}\|^2(1-\varepsilon_1)\tau\|g_n\|^2}\right|$$

$$\le \frac{(\|s_n\|.\|g_{n+1}\| + \|y_n\|.\|g_{n+1}\|).\|d_n\|^2 + \rho_n L\|g_{n+1}\|.\|y_n\|.\|d_n\|^2}{0.2(1-\varepsilon_1)\tau\bar{e}^2\bar{u}^2}$$

$$\le \frac{(1+L)uAD^2 + \rho_n l^2 AD^2}{0.2\,(1-\varepsilon_1)\tau\bar{u}^2\bar{e}^2} = G_3$$

$$\therefore |\theta_n| \le G_1$$

$$\|d_{n+1}\| \le \|g_{n+1}\| + [(1-G_3)G_1 + G_3G_2].\|d_n\|$$

$$\le u + GB = \varphi$$

$$\Rightarrow \sum_{n\ge1}\frac{1}{\|d_n\|^2} \ge \frac{1}{\varphi_1^2}\sum_{n\ge1} 1 = \infty$$

$$\Rightarrow lim_{n\to\infty}\, inf\|g_n\| = 0.$$

5. NUMERICAL EXPERIMENTS

In this section we selected some of test functions from CUTE [27] library, along with other large scale optimization problems presented in Andrei [28] and Bongartz [29]. All codes are written in double precision FORTRAN Language. And compiled Visual F90 (default compiler settings) on a Workstation Intel Pentium 4. The value of ρ_n is always compute by cubic fitting procedure.

We selecte (24) large scale unconstrained optimization problems in the extended or generalized form. Each problem was tested three times for a gradually increasing number of variables: $N = 1000, 5000\ and\ 10000$, all algorithms implemented the SWP (3) and (4) conditions with $\varepsilon_1 = 0.001$ and $\varepsilon_2 = 0.9$ and the stopping criterion $\|g_n\| \le 10^{-6}$ is used.

In some cases, the computation stopped due to the failure of the line search to find the positive step size, and thus it was considered as a failure denoted by (f). We record the number of iteration calls (ni), the number of function evaluations calls (nf), and the of test problems calls (N), for purpose of our comparisons. Table 1 gives the comparison depending in the ni and nf between β_n^{RMIL}, β_n^{MMWU} and the proposed method β_n^{HA}.

Table 2 gives the percentage performance of the proposed methods β_n^{HA} against β_n^{RMIL} and β_n^{MMWU}. We have seen that β_n^{RMIL} method saves (ni 9.94%), (nf 17.11%), and β_n^{HA} method saves (ni 53.42%), (nf 36.01%) compared with β_n^{MMWU} method. While Figure 1 gives the comparison between β_n^{RMIL}, β_n^{MMWU} and β_n^{HA}, using Well-known EX-Wood test function.

Table 1. A list of different test functions with SWP conditions

N	Test Function	Dimension	β^{RMIL}		β^{MMWU}		β^{HA}	
			ni	nf	ni	nf	ni	nf
1	CUBIC	1000	16	44	16	45	15	43
		5000	16	44	16	45	15	43
		10000	16	44	16	45	15	43
2	DIGONAL6	1000	2	7	2	7	2	7
		5000	3	9	3	9	3	9
		10000	3	9	3	9	3	9
3	DENSCHNB	1000	6	15	6	15	6	15
		5000	6	15	6	15	6	15
		10000	6	15	6	15	6	15
4	DENSCHNF	1000	14	30	12	26	17	39
		5000	14	30	13	28	18	41
		10000	15	32	13	28	18	41
5	DIXMAAN A	1000	6	15	6	15	5	13
		5000	6	15	6	15	6	15
		10000	5	13	5	13	5	13
6	DQDRTIC	1000	32	65	32	65	19	39
		5000	31	63	32	65	20	41
		10000	31	63	32	65	19	39
7	EXTENDED BEALE (MATRIXROM)	1000	11	29	12	29	10	22
		5000	11	29	12	29	10	22
		10000	11	29	12	29	10	22
8	EX BLOCK DIAGONAL BD1	1000	24	50	22	46	17	37
		5000	24	50	22	46	17	37
		10000	24	50	23	48	18	39
9	EXTENDED CLIFF	1000	6	29	6	29	6	29
		5000	6	29	6	29	6	29
		10000	6	29	6	29	6	29
10	EX FREUDENSTEIN & ROTH	1000	8	21	8	21	8	20
		5000	8	21	8	21	8	20
		10000	8	21	8	21	8	20
11	EA PENALITY	1000	7	22	8	159	7	21
		5000	8	23	8	23	8	23
		10000	10	35	10	34	10	35
12	EX WOOD	1000	217	441	248	503	103	212
		5000	147	301	210	427	102	210
		10000	161	330	207	421	156	318
13	EX ROSEN	1000	29	76	27	69	24	62
		5000	29	76	27	69	24	62
		10000	29	76	28	72	24	62
14	EX WHITE & HOLST	1000	16	44	16	45	15	43
		5000	16	44	16	45	15	43
		10000	16	44	16	45	15	43
15	EX HIMMELBAU	1000	24	251	26	276	24	251
		5000	8	394	8	1138	8	384
		10000	8	386	8	390	8	391
16	FRED	1000	9	23	10	27	9	24
		5000	10	25	10	27	9	24
		10000	10	25	f	f	9	24
17	GCANTREL	1000	47	497	52	462	43	414
		5000	54	614	57	546	48	494
		10000	57	664	61	616	51	546
18	HELICAL	1000	81	167	65	134	55	113
		5000	85	175	68	140	59	121
		10000	85	175	68	140	59	121
19	MIELE	1000	133	480	134	510	121	437
		5000	157	598	141	549	121	437
		10000	161	620	145	569	129	483
20	POWELL 3	1000	25	54	31	66	19	41
		5000	26	65	32	68	19	41
		10000	f	f	32	68	19	41
21	POWLL 4	1000	f	f	f	f	93	239
		5000	f	f	f	f	114	328
		10000	f	f	f	f	114	328
22	QURATIC	1000	1	4	1	4	1	4
		5000	1	4	1	4	1	4
		10000	1	4	1	4	1	4
23	ROSEN	1000	29	76	27	69	35	86
		5000	29	76	27	69	35	86
		10000	29	76	28	72	35	86
24	WOOD	1000	104	215	204	415	91	200
		5000	105	217	266	539	91	199
		10000	110	227	246	499	91	200

Table 2. The percentage performance of the proposed methods

Measures	β_n^{RMIL}	β_n^{MMWU}	β_n^{HA}
ni	90.06%	100%	46.58%
nf	82.89%	100%	63.99%

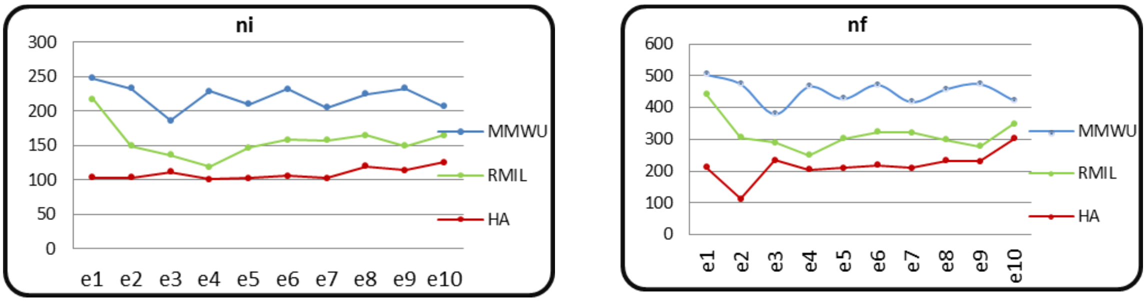

Figure 1. The compare between three methods

6. CONCLUSION

In this paper, a new parameter θ_n for a hybrid conjugate gradient is derived. The practical results indicated that the proposed hybrid method is faster and more efficient compared to the β_n^{RMIL} and β_n^{MMWU} algorithms used.

ACKNOWLEDGEMENTS

The authors are very grateful to the University of Mosul, College of Computer Science and Mathematics for their provided facilities, which helped to improve the quality of this work.

REFERENCES

[1] J. K. Liu and S. J. Li, "New hybrid conjugate gradient method for unconstrained optimization," *Applied Mathematics and Computation*, vol. 245, pp. 36-43, 2014.
[2] Z. Salleh and A. Alhawarat, "An efficient modification of the Hestenes-Stiefel nonlinear conjugate gradient method with restart property," *Journal of Inequalities and Applications*, vol. 110, 2016.
[3] P. Wolfe, "Convergence conditions for ascent methods," *SIAM review*, vol. 11, no. 2, pp. 226-235, 1969.
[4] P. Wolfe, "Convergence Conditions for Ascent Methods. II: Some Corrections," *SIAM Review*, vol. 13, no. 2, pp. 185-188, 1971.
[5] A. Alhawarat, et al., "An efficient hybrid conjugate gradient method with the strong Wolfe-Powell line search," *Mathematical Problems in Engineering*, vol. 2015, no. 14, pp. 1-7, 2015.
[6] C. Li, "A Modified Conjugate Gradient Method for Unconstrained Optimization," *TELKOMNIKA*, vol. 11, no. 11, pp. 6373-6380, 2013.
[7] M. R. Hestenes and E. Stiefel, "Methods of conjugate gradients for solving linear systems," *Journal of research of the National Bureau of Standards*, vol. 49, no. 6, pp. 409-436, 1952.
[8] R. Fletcher and C. M. Reeves, "Function minimization by conjugate gradients," *The computer journal*, vol. 7, no. 2, pp. 149-154, 1964.
[9] E. Polak and G. Ribiere, "Note sur la convergence de méthodes de directions conjuguées," *ESAIM: Mathematical Modelling and Numerical Analysis-Modélisation Mathématique et Analyse Numérique*, vol. 3, no. R1, pp. 35-43, 1969.
[10] B. T. Polyak, "The conjugate gradient method in extreme problems," *USSR Computational Mathematics and Mathematical Physics*, vol. 9, no. 4, pp. 94-112, 1969.
[11] R. Fletcher, "Practical methods of optimization," John Wiley & Sons, 1987.
[12] Y. Liu and C. Storey, "Efficient generalized conjugate gradient algorithms, part 1: theory," *Journal of optimization theory and applications*, vol. 69, no. 1, pp. 129-137, 1991.
[13] Y. H. Dai and Y. Yuan, "A nonlinear conjugate gradient method with a strong global convergence property," *SIAM Journal on optimization*, vol. 10, no. 1, pp. 177-182, 1999.
[14] G. M. Al-Naemi and E. T. Hamed, "New conjugate gradient method with Wolfe type line searches for nonlinear programming," *Australian Journal of Basic and Applied Sciences*, vol. 7, no. 14, pp. 622-632, 2013.
[15] N. S. Mohamed, et al., "A New Hybrid Coefficient of Conjugate Gradient method," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 18, no. 3, pp. 1454-1463, 2020.
[16] N. Andrei, "Another nonlinear conjugate gradient algorithm for unconstrained optimization," *Optimization methods & software*, vol. 24, no. 1, pp. 89-104, 2009.

[17] S. S. Djordjevic, "New Hybrid Conjugate Gradient Method as a Convex Combination of LS and CD Methods," *Filomat*, vol. 31, no. 6, pp. 1813-1825, 2017.
[18] X. Zheng, et al., "Further comment on another hybrid conjugate gradienyt algorithm for unconstrained optimization," *Numerical Algorithms*, vol. 84, pp. 603-608, 2020.
[19] I. E. Livieris, et al., "A descent hybrid conjugate gradient method based on the memoryless BFGS update," *Numerical Algorithms*, vol. 79, no. 4, pp. 1169-1185, 2018.
[20] F. N. Al-Namat and G. M. Al-Naemi, "Global convergence property with inexact line search for a new hybrid conjugate gradient method," *Open Access Library Journal*, vol. 7, no. 2, pp. 1-15, 2020.
[21] R. B. Yunus, et al., "Comparative Study of some New Conjugate Gradient Methods," in *UniSZA Research Conference*, 2015.
[22] A. V. Mandara, et al., "A New Conjugate Gradient Coefficient With Exact Line Search for unconstrained Optimization," *Far East Journal of Mathematical Sciences (FJMS)*, vol. 105, no. 2, pp. 193-206, 2018.
[23] B. A. Hassan and G. M. Al-Naemi, "A New Quasi-Newton Equation on the Gradient Methods for Optimization Minimization Problem to appear," in *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 19, no. 2, pp. 737-744, 2020.
[24] 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.
[25] G. M. Al-Naemi, "Modified Nonlinear CG Algorithms with Application in Neural Networks," LAP Lambert Academic Publishing, Saarbrucken, 2013.
[26] N. S. Mohamed, et al., "Global Convergence of a New Coefficient Conjugate Gradient Method," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 11, no. 3, pp. 1188-1193, 2018.
[27] N. Andrei, "An Unconstrained Optimization Test Functions Collection," *Advanced Modellling and Optimization*, vol. 10, no. 1, pp. 147-161, 2008.
[28] N. Andrei, "Test functions for unconstrained optimization," Research Institute for informatics, Center for Advanced Modeling and Optimization, pp. 1-15, 2004.
[29] I. Bongartz, et al., "CUTE: Constrained and Unconstrained Testing Environment," *ACM Transactions on Mathematical Software*, vol. 21, no. 1, pp. 123-160, 1995.