Annals of Mathematics and Computer Science
ISSN: 2789-7206
Vol 18 (2023) 26-39

TWO NEW CONJUGATE GRADIENT METHODS IN UNCONSTRAINED OPTIMIZATION PROBLEMS

AYINDE, SEMIU A.[1]*, AGBOOLA SUNDAY O.[2], ADELODUN, JOSEPH F. [3] AND HASSAN, SAMUEL O.[4]

Abstract. Zheng Y. and Zheng B. modified Dai-Liao conjugate gradient method to come up with two new Dai-Liao-type conjugate gradient methods. These methods were shown to have satisfied descent condition taken into consideration the strong Wolfe line search. Convergence for objective functions were also guarantied. In this work, two new conjugate gradient methods are introduced in line with the work of Zheng Y. and Zheng B. by changing the first term in AyO-CG method to solve unconstrained non-linear optimization problems. Descent properties of these methods are shown and guarantied. Convergence analysis of these methods in line with strong Wolfe conditions showed that they are globally convergent. Comparison based on Dolan More performance profile of the numerical strength of these methods with the two modified Dai-Laio type methods proved that our methods outperformed them.

1. Introduction and preliminaries

Many authors have worked on conjugate gradient methods (CG methods) as a result of their wide acceptability in solving unconstrained optimization problems and also because of their global convergence and low memory status. It is worthy of note that most of the proposed CG methods are modifications of the first generation CG methods which are also known as classical CG methods. For details on this see [1, 5, 13, 15]. Notable among these, is the work of Dai and Liao [4] tagged DL CG method, where they considered an inexact line search conjugacy scheme that reduces to a classical exact line search conjugacy scheme to solve non-linear unconstrained optimization problem. They gave global convergence properties of the method for general functions. It is also worthy of note that a number of improvements on this work have also been presented which include: Yabe and Takano [11] and Li *et al*. [9] modified the secant equation on the Dai-Liao method; Lu [7] presented a modified Dai-Liao conjugacy condition based on a new quasi-Newton equation; Zheng [15] also introduced a secant equation into Dai-Liao conjugacy condition. Other work in this direction in improving Dai-Liao

---

Date: Received: Aug 17, 2023; Accepted: Sep 12.

* Corresponding author.

2010 *Mathematics Subject Classification.* Primary 49J52, 49J53; Secondary 90C31.

Key words and phrases. Global convergence; unconstrained optimization; strong Wolfe conditions; descent direction; step length.

method can be found in [7, 12, 13]. Of interest is the work of Zheng Y. and Zheng B. [14] that presented two new Dai-Liao type CG methods by replacing the first term of

$$\beta_k^{DL} = max\{\frac{g_k^T y_{k-1}}{y_{k-1}^T d_{k-1}}, 0\} - t\frac{g_k^T s_{k-1}}{d_{k-1}^T y_{k-1}} \tag{1.1}$$

with

$$\beta_k^{DHS} = \frac{||g_k||^2 - \frac{||g_k||}{||g_{k-1}||}|d_{k-1}^T g_k|}{\mu|d_{k-1}^T g_k| + y_{k-1}^T d_{k-1}} \tag{1.2}$$

and

$$\beta_k^{DLS} = \frac{||g_k||^2 - \frac{||g_k||}{||g_{k-1}||}|d_{k-1}^T g_k|}{\mu|d_{k-1}^T g_k| - d_{k-1}^T g_{k-1}} \tag{1.3}$$

respectively, to give

$$\beta_k^{DHSDL} = \frac{||g_k||^2 - \frac{||g_k||}{||g_{k-1}||}|d_{k-1}^T g_k|}{\mu|d_{k-1}^T g_k| + y_{k-1}^T d_{k-1}} - t\frac{g_k^T s_{k-1}}{d_{k-1}^T y_{k-1}} \tag{1.4}$$

and

$$\beta_k^{DLSDL} = \frac{||g_k||^2 - \frac{||g_k||}{||g_{k-1}||}|d_{k-1}^T g_k|}{\mu|d_{k-1}^T g_k| - d_{k-1}^T g_{k-1}} - t\frac{g_k^T s_{k-1}}{d_{k-1}^T y_{k-1}} \tag{1.5}$$

respectively. They considered their global convergence alongside their numerical strength and were found to have outperformed other types of Dai-Liao methods. Ayinde *et al* [3] developed what they called a new Dai-Liao type CG method by constructing new β_k towing the line of inexact line scheme to come up with a CG method that uses

$$\beta_k^{AyO} = \frac{||g_k||^2}{d_{k-1}^T y_{k-1}} + \frac{t g_k^T s_{k-1}}{d_{k-1}^T g_{k-1}}. \tag{1.6}$$

CG methods with conjugate gradient coefficient β_k comprising linear combination of terms have been shown to be more efficient and promising. This is also evident from the work of Zheng Y. and Zheng B. [14]. Our motivation comes form the work of Zheng Y. and Zheng B. [14] coupled with the fact that AyO CG method was shown to have performed better that the DL CG method. See [3]. In this work, we present two new CG methods by working in line with the work presented in [14]. We showed the convergence of our methods. Experiments performed showed that numerical strength of our proposed methods outwit the two CG methods presented in [14].

In section 2, the two new methods are presented. Analysis on convergence is done in section 3. We consider the numerical experiment and discussion of results in section 4 and section 5 gives the conclusion.

2. New CG Methods

Given an unconstrained optimization problem in n variables

$$f(u),\ u \in R^n \tag{2.1}$$

with f being continuous and differentiable, the aim is to minimize (2.1).
The CG methods have been found suitable for solving equation of the form (2.1) using the iterative formula stated as follows

$$u_{k+1} = u_k + \alpha_k d_k, \tag{2.2}$$

where the step-length $\alpha_k > 0$ is to be determined by a line search schemes with the direction d_k given as

$$d_k = \begin{cases} -g_k, & k = 0, \\ -g_k + \beta_k d_{k-1}, & k \geq 1. \end{cases} \tag{2.3}$$

The formula $\beta_k \in R$ is a scalar that must satisfy the condition $d_k^T y_{k-1} = 0$ with $y_{k-1} = g_k - g_{k-1}$. The formula β_k determines the nature of the CG-method.
In all, the choice of the step-length α_k is a factor for global convergence. Hence, the following strong Wolfe conditions

$$f(u_k) - f(u_k + \alpha_k d_k) \geq -\delta \alpha_k d_k^T g_k \tag{2.4}$$

and

$$g(u_k + \alpha_k d_k)^T d_k \geq \sigma d_k^T g_k \tag{2.5}$$

for $0 < \delta \leq \sigma < 1$ are needed to estimate α_k. g_k represents gradient of $f(x_k)$.
Following the approach adopted in (1.1) to (1.5), this paper proposes the following conjugate gradient coefficients for the two new conjugate gradient methods by replacing the first term of (1.6) with

$$\beta_k^{DHS} = \frac{||g_k||^2 - \frac{||g_k||}{||g_{k-1}||}|d_{k-1}^T g_k|}{\mu|d_{k-1}^T g_k| + y_{k-1}^T d_{k-1}} \tag{2.6}$$

and

$$\beta_k^{DLS} = \frac{||g_k||^2 - \frac{||g_k||}{||g_{k-1}||}|d_{k-1}^T g_k|}{\mu|d_{k-1}^T g_k| - d_{k-1}^T g_{k-1}} \tag{2.7}$$

respectively, to give

$$\beta_k^{DHSAyO} = \frac{||g_k||^2 - \frac{||g_k||}{||g_{k-1}||}|d_{k-1}^T g_k|}{\mu|d_{k-1}^T g_k| + y_{k-1}^T d_{k-1}} + \frac{t s_{k-1}^T g_k}{(d_{k-1}^T g_{k-1})} \tag{2.8}$$

and

$$\beta_k^{DLSAyO} = \frac{||g_k||^2 - \frac{||g_k||}{||g_{k-1}||}|d_{k-1}^T g_k|}{\mu|d_{k-1}^T g_k| - d_{k-1}^T g_{k-1}} + \frac{t s_{k-1}^T g_k}{(d_{k-1}^T g_{k-1})} \tag{2.9}$$

respectively, whereby $\mu \geq 1$, $t > 0$ and $s_{k-1} = u_k - u_{k-1}$.

Procedure (2.2) -(2.3) with $\beta_k^{DHSDL}(\beta_k^{DLSDL})$ is called DHSDL (DLSDL) method.
Procedure (2.2) -(2.3) with β_k^{DHSAyO} (β_k^{DLSAyO}) is called DHSAyO (DLSAyO) method.

2.1. **Algorithm for the Method.** Step 1: With the initial point $u_o \in R^n$, we set $k = 1$ and $d_0 = -g_0$. If $||g_k|| \leq \epsilon$ where $\epsilon > 0$ stop.
Step 2: Find $\alpha_k > 0$ by conditions (2.4) to (2.5).
Step 3: Compute $\beta_k^{DLSAyO}(\beta_k^{DHSAyO})$ and determine the sequences $\{u_k\}$, $\{g_k\}$ and $\{d_k\}$
Step 4: Put $k = k + 1$ and go to step 2.

3. Convergence of the methods

In this section, discussions on the convergence of the proposed methods are considered. We shall consider convergence analysis of DLSAyO method only, since the same procedures can be used for DHSAyO method.

Assumption 3.1
The objective function (2.1) satisfies the conditions highlighted below.
(1) The set

$$U = \{u | f(u) \leq f(v)\} \tag{3.1}$$

with $v \in R^n$ is bounded.
(2) f is continuous and differentiable in W, a neighborhood in U and its gradient $g(u) = \triangledown f(u)$ with Lipschitz constant L satisfies Lipschitz continuity

$$||\triangledown f(u) - \triangledown f(v)|| \leq L||u - v|| \tag{3.2}$$

for any $u, v \in W$.

Furthermore, by Assumption 3.1 there exists two constants D and ϕ that satisfy the following inequalities

$$||u - v|| \leq D \tag{3.3}$$

for any $u, v \in W$
and

$$||\triangledown f(u)|| \leq \phi \tag{3.4}$$

for any $u \in W$.

Lemma 3.1. *Suppose d_k and g_k are determine by the CG method algorithm given in section 2. Then, d_k satisfied the following condition*

$$g_k^T d_k \leq -c||g_k||^2 \tag{3.5}$$

for each $k \geq 0$.

Proof. We show by induction. It is obvious for the case when $k = 0$ that

$$g_0^T d_0 = -||g_0||^2. \tag{3.6}$$

Let (3.5) be valid for $k \geq 1$. From (2.3), we have

$$d_k = -g_k + \beta_k d_{k-1}. \tag{3.7}$$

By pre-multiplying (3.7) by g_k,

$$g_k^T d_k = -||g_k||^2 + \beta_k^{DLSAyO} g_k^T d_{k-1}. \tag{3.8}$$

By using (3.8) coupled with $s_{k-1} = \alpha_{k-1}d_{k-1}$, we have

$$g_k^T d_k = -||g_k||^2 + \beta_k^{DLS} g_k^T d_{k-1} + \frac{t\alpha_{k-1}d_{k-1}^T g_k(g_k^T d_{k-1})}{(d_{k-1}^T g_{k-1})}. \tag{3.9}$$

Since

$$d_{k-1}^T y_{k-1} = (g_k^T - g_{k-1}^T)d_{k-1} \tag{3.10}$$

$$= d_{k-1}g_k^T - d_{k-1}g_{k-1}^T \tag{3.11}$$

$$\leq |d_{k-1}g_k^T| - d_{k-1}g_{k-1}^T. \tag{3.12}$$

For (3.12) and $d_{k-1}^T y_{k-1} > 0$ to hold always, it implies that

$$-d_{k-1}g_{k-1}^T > 0. \tag{3.13}$$

i.e

$$d_{k-1}g_{k-1}^T < 0. \tag{3.14}$$

So also, by [10], $\beta_k^{DLS} \leq \frac{||g_k||^2}{\mu|d_{k-1}^T g_k|}$. Therefore,

$$g_k^T d_k \leq -||g_k||^2 + \beta_k^{DLS} g_k^T d_{k-1} \tag{3.15}$$

$$\leq -||g_k||^2 + \frac{||g_k||^2}{\mu|d_{k-1}^T g_k|} g_k^T d_{k-1} \tag{3.16}$$

$$\leq -||g_k||^2 + \frac{||g_k||^2}{\mu}. \tag{3.17}$$

Hence,

$$g_k^T d_k \leq -(1 - \frac{1}{\mu})||g_k||^2. \tag{3.18}$$

For $\lambda \geq 1$, we have the above inequality being satisfied where $c = (1 - \frac{1}{\mu})$.

□

Theorem 3.2. *Suppose the conditions in the assumption 3.1 hold and also that u_k be determined by the algorithm in section 2. Then,*

$$\lim_{k\to\infty} \inf ||g_k|| = 0. \tag{3.19}$$

Proof. Going by (3.14), we have

$$\beta_k^{DLSAyO} = \frac{||g_k||^2 - \frac{||g_k||}{||g_{k-1}||}|d_{k-1}^T g_k|}{\mu|g_k d_{k-1}^T| - d_{k-1}^T g_{k-1}} + \frac{t s_{k-1}^T g_k}{(d_{k-1}^T g_{k-1})} \tag{3.20}$$

$$\leq \frac{||g_k||^2 - \frac{||g_k||}{||g_{k-1}||}|d_{k-1}^T g_k|}{-d_{k-1}^T g_{k-1}} + \frac{t s_{k-1}^T g_k}{(d_{k-1}^T g_{k-1})} \tag{3.21}$$

$$\leq \left|\frac{||g_k||^2 - \frac{||g_k||}{||g_{k-1}||}|d_{k-1}^T g_k|}{-d_{k-1}^T g_{k-1}} + \frac{t s_{k-1}^T g_k}{(d_{k-1}^T g_{k-1})}\right| \tag{3.22}$$

$$= \left|\frac{||g_k||^2 - \frac{||g_k||}{||g_{k-1}||}|d_{k-1}^T g_k|}{-d_{k-1}^T g_{k-1}} - \frac{t s_{k-1}^T g_k}{(-d_{k-1}^T g_{k-1})}\right| \tag{3.23}$$

$$\leq |\frac{||g_k||^2 - \frac{||g_k||}{||g_{k-1}||}|d_{k-1}^T g_k|}{-d_{k-1}^T g_{k-1}} + \frac{t s_{k-1}^T g_k}{(-d_{k-1}^T g_{k-1})}| \tag{3.24}$$

$$\leq \frac{||g_k||(||g_k - g_{k-1}|| + ||g_k - g_{k-1}||) + t||g_k||||s_{k-1}||}{|(-g_{k-1}^T d_{k-1})|} \tag{3.25}$$

$$\leq \frac{2||g_k||||g_k - g_{k-1}|| + t||g_k||||s_{k-1}||}{|(-g_{k-1}^T d_{k-1})|}. \tag{3.26}$$

By Lemma 3.1

$$\beta_k^{DLSAyO} \leq \frac{2L||g_k||||s_{k-1}|| + t||g_k||||s_{k-1}||}{c||g_{k-1}||^2} \tag{3.27}$$

$$\leq \frac{(2L+t)||g_k||||s_{k-1}||}{c||g_{k-1}||^2}. \tag{3.28}$$

By ([10], Theorem 5.8), $||g_k|| \geq c_1$ for all k therefore,

$$\beta_k^{DLSAyO} \leq \frac{(2L+t)||s_{k-1}||||g_k||}{cc_1^2}. \tag{3.29}$$

$$From(3.7),\ ||d_k|| \leq ||g_k|| + |\beta_k^{DLSAyO}|||d_{k-1}||. \tag{3.30}$$

$$||d_k|| \leq ||g_k|| + \frac{(2L+t)||s_{k-1}||||g_k||}{cc_1^2}||d_{k-1}||. \tag{3.31}$$

$$Hence,\ ||d_k|| \leq ||g_k|| + \frac{(2L+t)||s_{k-1}||||g_k||}{cc_1^2}||s_{k-1}||. \tag{3.32}$$

Since $s_{k-1} = u_k - u_{k-1}$ coupled with (3.3), we then have

$$||d_k|| \leq ||g_k|| + \frac{((2L+t)D)D||g_k||}{cc_1^2}. \tag{3.33}$$

$$||d_k|| \leq (1 + \frac{((2L+t)D^2}{cc_1^2})||g_k||. \tag{3.34}$$

By using (3.5) and (3.34) together with Zoutendijk's condition in [16], we have

$$\sum_{k\geq 1}(1-\frac{1}{\mu})^2(1+\frac{((2L+t)D^2}{cc_1^2})^{-2}||g_k||^2 \leq \sum_{k\geq 1}\frac{(g_k^T d_k)^2}{||d_k||^2} < +\infty. \tag{3.35}$$

Inequality (3.35) obviously implies the result. □

4. Numerical Experiment and Discussion of Results

A number of test functions taken from [2] are considered here to determine the numerical strength of the proposed methods in comparison with other methods.

4.1. **Numerical Experiment.** This subsection reports the numerical strength of DLSAyO and DHSAyO methods against DLSDL and DHSDL methods considered in [14].

Table 1. Numerical result of IT and F

s/n	$Prob.$	$Dim.$	$DLSAyO$	$DLSDL$
			IT/F	IT/F
1	$Ext.Penalty$	500	$1729/2.49E+05$	$1805/2.49E+05$
2		1000	$3007/9.97E+05$	NAN/NAN
3	$Rosenbrock$	500	$112/7.35E-15$	$151/8.76E-14$
4		1000	$145/1.10E-14$	$162/1.42E-13$
5		10000	$169/1.16E-15$	$146/2.52E-14$
6	$Ext.Rosenbrock$	500	$102/5.70E-16$	$201/5.54E-16$
7		1000	$82/2.51E-16$	$147/2.02E-16$
8		10000	$102/5.24E-16$	$193/3.68E-16$
9	$Diagonal4$	500	$15/7.82E-17$	$15/7.86E-17$
10		1000	$15/1.56E-16$	$15/1.57E-16$
11		10000	$15/1.56E-15$	$15/1.57E-15$
12	$Ext.Himmelblau$	500	$28/4.33E-16$	$36/3.84E-17$
13		1000	$28/8.67E-16$	$36/8.80E-17$
14		10000	$29/5.80E-16$	$36/1.15E-15$
15	$Raydan1$	500	$1/8.73E+07$	$1/8.73E+07$
16		1000	$1/1.39E+09$	$1/1.39E+09$
17		10000	$1/1.39E+13$	$1/1.39E+13$
18	$Diagonal1$	500	$6/5.00E+02$	$6/5.00E+02$
19		1000	$6/1.00E+03$	$6/1.00E+03$
20		10000	$6/1.00E+04$	$6/1.00E+04$
21	$Ext.Tridiagonal1$	500	$4366/6.15E-09$	$4383/5.92E-09$
22		1000	$5564/7.74E-09$	$5716/7.57E-09$
23		10000	$11892/1.66E-08$	NAN/NAN
24	$Ext.3Exp.Terms$	500	$23/6.40E+02$	$26/6.40E+02$
25	$Gen.Rosenbrock$	500	$388/9.94E-14$	$468/4.89E-14$
26	$Ext.BD1$	500	$32/1.11E-14$	$36/4.80E-21$
27		1000	$32/2.22E-14$	$36/9.59E-21$
28		10000	$36/1.59E-17$	$36/9.59E-20$
29	$Ext.Cliff$	500	$438/4.99E+01$	$605/4.99E+01$
30		1000	$325/9.99E+01$	$553/9.99E+01$
31		10000	$2262/9.99E+02$	$1090/9.99E+02$
32	$Ext.Wood$	500	$656/3.89E-15$	$502/1.09E-13$
33		1000	$173/NAN$	$844/2.46E-13$
34		10000	$983/1.43E-13$	$662/6.43E-14$
35	$FLETCHCR$	500	$5070/1.70E-11$	$4850/4.95E-12$
36		1000	$7966/9.18E-11$	$7764/3.36E-12$
37	$ARWHEAD$	500	$1729/2.49E+05$	$1805/2.49E+05$
38		1000	$3007/9.97E+05$	NAN/NAN
39	$DQDRTIC$	500	$104/1.86E-14$	$82/1.64E-15$
40		1000	$58/1.61E-14$	$91/9.03E-14$
41		10000	$113/5.41E-14$	$82/3.85E-15$

TABLE 2. Numerical result of IT and F contd.

s/n	$Prob.$	$Dim.$	$DHSDL$	$DHSAyO$
			IT/F	IT/F
1	$Ext.Penalty$	500	$175/2.49E+05$	$141/2.49E+05$
2		1000	$3335/9.97E+05$	$3293/9.97E+05$
3	$Rosenbrock$	500	$265/1.11E-15$	$154/6.35E-14$
4		1000	$166/2.90E-13$	$187/1.61E-13$
5		10000	$199/9.96E-15$	$163/2.55E-15$
6	$Ext.Rosenbrock$	500	$143/3.90E-16$	$96/4.61E-17$
7		1000	$265/1.60E-16$	$164/3.44E-16$
8		10000	$139/2.54E-16$	$269/4.53E-16$
9	$Diagonal4$	500	$23/2.47E-16$	$23/2.48E-16$
10		1000	$23/4.95E-16$	$23/4.96E-16$
11		10000	$24/7.38E-16$	$24/7.40E-16$
12	$Ext.Himmelblau$	500	$20/1.40E-15$	$21/1.36E-15$
13		1000	$20/2.79E-15$	$21/2.71E-15$
14		10000	$21/1.87E-15$	$22/1.81E-15$
15	$Raydan1$	500	$1/8.73E+07$	$1/8.73E+07$
16		1000	$1/1.39E+09$	$1/1.39E+09$
17		10000	$1/1.39E+13$	$1/1.39E+13$
18	$Diagonal1$	500	$6/5.00E+02$	$6/5.00E+02$
19		1000	$6/1.00E+03$	$6/1.00E+03$
20		10000	$6/1.00E+04$	$6/1.00E+04$
21	$Ext.Tridiagonal1$	500	NAN/NAN	$6016/6.09E-09$
22		1000	NAN/NAN	$7666/7.45E-09$
23		10000	NAN/NAN	$16289/1.65E-08$
24	$Ext.3Exp.Terms$	500	$28/6.40E+02$	$25/6.40E+02$
25	$Gen.Rosenbrock$	500	$735/4.68E-13$	$1362/3.50E-13$
26	$Ext.BD1$	500	$42/2.34E-17$	$50/3.87E-15$
27		1000	$42/4.68E-17$	$50/9.60E-15$
28		10000	$42/4.68E-16$	$50/8.73E-14$
29	$Ext.Cliff$	500	$5795/4.99E+01$	$5236/4.99E+01$
30		1000	$4490/9.99E+01$	$7689/9.99E+01$
31		10000	$4109/9.99E+02$	$53178/9.99E+02$
32	$Ext.Wood$	500	$971/3.13E-13$	$1109/2.73E-13$
33		1000	$1531/7.62E-14$	$1228/1.50E-13$
34		10000	$1470/8.27E-14$	$1146/2.41E-13$
35	$FLETCHCR$	500	$5110/5.45E-12$	$7485/1.89E-11$
36		1000	$22400/1.03E-10$	$16300/1.06E-10$
37	$ARWHEAD$	500	$175/2.49E+05$	$141/2.49E+05$
38		1000	$3340/9.97E+05$	$3290/9.97E+05$
39	$DQDRTIC$	500	$94/1.34E-14$	$110/4.62E-14$
40		1000	$86/7.27E-14$	$86/4.19E-15$
41		10000	$107/3.16E-14$	$106/1.14E-13$

TABLE 3. Numerical result of CPU time and gradient norm.

s/n	$Prob.$	$Dim.$	$DLSAyO$	$DLSDL$
			G/CPU	G/CPU
1	$Ext.Penalty$	500	$1.58E-7/16.899$	$9.95E-07/17.21$
2		1000	$5.02E-7/32.695$	NAN/NAN
3	$Rosenbrock$	500	$7.28E-7/1.004$	$6.77E-07/1.276$
4		1000	$9.00E-7/1.36$	$6.75E-07/1.418$
5		10000	$9.39E-7/2.963$	$7.38E-07/2.624$
6	$Ext.Rosenbrock$	500	$9.56E-7/0.869$	$9.65E-07/2.185$
7		1000	$8.11E-7/0.78$	$5.68E-07/1.453$
8		10000	$9.37E-7/2.283$	$9.22E-07/2.318$
9	$Diagonal4$	500	$2.18E-7/0.131$	$2.18E-07/0.141$
10		1000	$3.08E-7/0.15$	$3.08E-07/0.148$
11		10000	$9.73E-7/0.341$	$9.75E-07/0.344$
12	$Ext.Himmelblau$	500	$2.64E-7/0.506$	$9.10E-08/0.355$
13		1000	$3.74E-7/0.263$	$1.40E-7/0.405$
14		10000	$3.06E-7/0.713$	$5.20E-07/0.923$
15	$Raydan1$	500	$0.00E+0/0.011$	$0.00E+00/0.012$
16		1000	$0.00E+0/0.012$	$0.00E+00/0.012$
17		10000	$0.00E+0/0.013$	$0.00E+00/0.032$
18	$Diagonal1$	500	$2.55E-09/0.049$	$2.36E-09/0.055$
19		1000	$3.61E-9/0.049$	$3.34E-09/0.057$
20		10000	$1.14E-8/0.65$	$1.06E-08/0.081$
21	$Ext.Tridiagonal1$	500	$9.96E-7/38.246$	$9.63E-07/38.46$
22		1000	$9.87E-7/50.692$	$9.71E-07/53.27$
23		10000	$9.85E-7/219.52$	NAN/NAN
24	$Ext.3Exp.Terms$	500	$8.93E-7/0.223$	$7.52E-07/0.253$
25	$Gen.Rosenbrock$	500	$7.08E-7/3.726$	$9.93E-07/4.327$
26	$Ext.BD1$	500	$5.96E-7/0.266$	$1.96E-10/0.301$
27		1000	$8.43E-7/0.276$	$2.78E-10/0.307$
28		10000	$1.13E-8/0.577$	$8.78E-10/0.591$
29	$Ext.Cliff$	500	$4.38E-7/3.712$	$8.69E-07/6.245$
30		1000	$7.66E-7/2.781$	$6.42E-07/4.90$
31		10000	$8.69E-7/44.432$	$9.24E-07/21.75$
32	$Ext.Wood$	500	$9.79E-7/6.134$	$6.92E-07/4.75$
33		1000	$NAN/1.734$	$8.77E-7/8.397$
34		10000	$7.66E-7/27.011$	$9.47E-07/18.8$
35	$FLETCHCR$	500	$9.86E-7/51.48$	$6.00E-07/55.5$
36		1000	$9.74E-7/94.673$	$9.84E-07/93.8$
37	$ARWHEAD$	500	$1.58E-7/17.608$	$9.95E-07/17.3$
38		1000	$5.02E-7/37.03$	NAN/NAN
39	$DQDRTIC$	500	$7.63E-7/0.869$	$7.26E-07/0.753$
40		1000	$8.42E-7/0.522$	$8.56E-07/0.815$
41		10000	$8.54E-7/1.931$	$8.47E-07/1.39$

TABLE 4. Numerical result of CPU time and gradient norm contd.

s/n	$Prob.$	$Dim.$	$DHSDL$	$DHSAyO$
			G/CPU	G/CPU
1	$Ext.Penalty$	500	$1.62E-07/1.652$	$8.21E-07/1.309$
2		1000	$4.62E-07/36.43$	$5.01E-08/35.75$
3	$Rosenbrock$	500	$7.19E-07/2.245$	$3.85E-07/1.274$
4		1000	$9.76E-07/1.536$	$7.87E-07/1.644$
5		10000	$9.41E-07/3.461$	$8.64E-07/2.96$
6	$Ext.Rosenbrock$	500	$9.01E-07/1.389$	$8.69E-07/0.89$
7		1000	$7.86E-07/2.868$	$9.89E-07/1.552$
8		10000	$9.18E-07/3.196$	$9.27E-07/5.989$
9	$Diagonal4$	500	$4.19E-07/0.24$	$4.19E-07/0.225$
10		1000	$5.92E-07/0.224$	$5.93E-07/0.514$
11		10000	$7.24E-07/507$	$7.24E-07/0.52$
12	$Ext.Himmelblau$	500	$4.74E-07/0.226$	$4.67E-07/0.201$
13		1000	$6.71E-07/0.214$	$6.61E-07/0.246$
14		10000	$5.49E-07/0.531$	$5.40E-07/0.548$
15	$Raydan1$	500	$0.00E+00/0.012$	$0.00E+00/0.012$
16		1000	$0.00E+00/0.028$	$0.00E+00/0.022$
17		10000	$0.00E+00/0.022$	$0.00E+00/0.018$
18	$Diagonal1$	500	$4.78E-10/0.068$	$1.51E-09/0.057$
19		1000	$6.76E-10/0.064$	$2.14E-09/0.051$
20		10000	$2.14E-09/0.112$	$6.77E-09/0.076$
21	$Ext.Tridiagonal1$	500	NAN/NAN	$9.88E-07/52.64$
22		1000	NAN/NAN	$9.68E-07/71.264$
23		10000	NAN/NAN	$9.86E-07/308.35$
24	$Ext.3Exp.Terms$	500	$6.27E-07/0.242$	$6.01E-07/0.235$
25	$Gen.Rosenbrock$	500	$8.89E-07/6.897$	$8.77E-07/12.532$
26	$Ext.BD1$	500	$1.44E-08/0.353$	$1.76E-07/0.414$
27		1000	$2.03E-08/0.361$	$2.77E-07/0.433$
28		10000	$6.42E-08/0.739$	$8.36E-07/0.81$
29	$Ext.Cliff$	500	$8.83E-07/58.376$	$9.87E-07/52.38$
30		1000	$6.05E-07/45.2$	$3.95E-07/80.729$
31		10000	$9.48E-07/87.543$	$9.50E-07/1016.1$
32	$Ext.Wood$	500	$9.96E-07/9.72$	$9.72E-07/11.555$
33		1000	$9.88E-07/17.358$	$7.61E-07/12.9$
34		10000	$8.89E-07/44.1$	$9.96E-07/33.8$
35	$FLETCHCR$	500	$8.79E-07/56.5$	$8.61E-07/92.12$
36		1000	$9.88E-07/280$	$9.69E-07/201$
37	$ARWHEAD$	500	$1.62E-07/1.58$	$8.21E-07/1.315$
38		1000	$4.62E-07/36.5$	$5.01E-08/42.8$
39	$DQDRTIC$	500	$9.06E-07/0.840$	$7.01E-07/0.974$
40		1000	$7.37E-07/0.768$	$9.25E-07/0.7.67$
40		10000	$8.30E-07/1.93$	$8.29E-07/1.83$

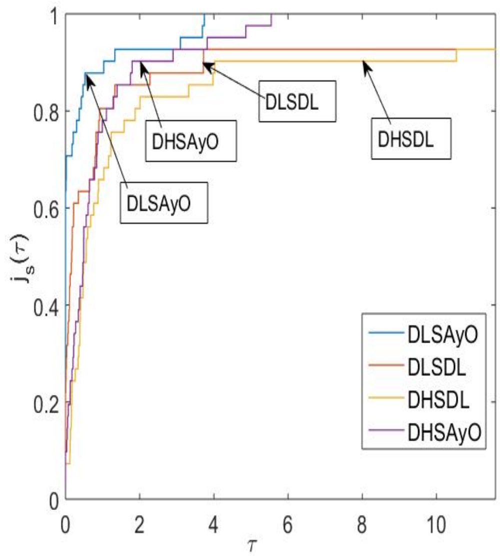

FIGURE 1. CPU time

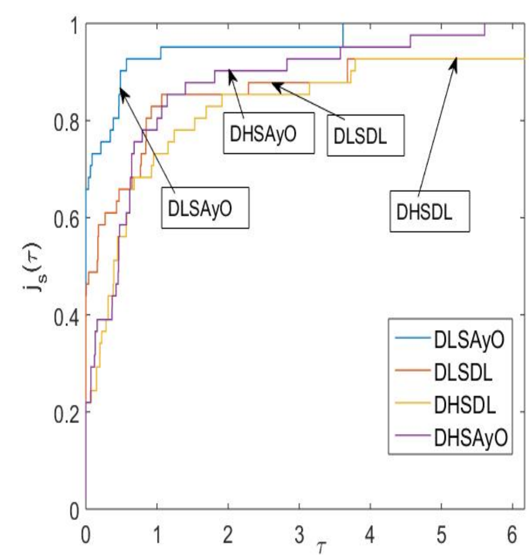

FIGURE 2. Number of iterations (ITR)

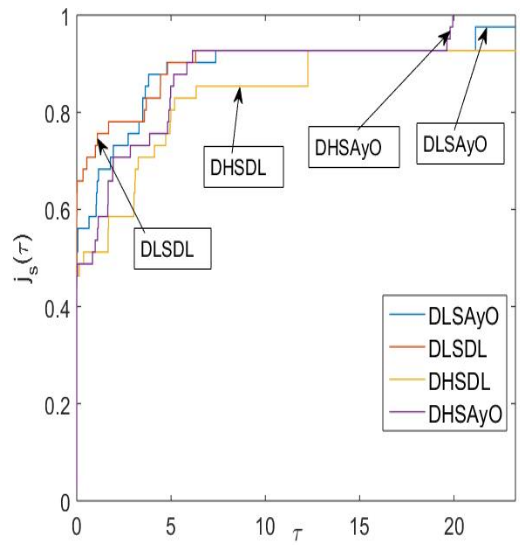

FIGURE 3. value of Function (F)

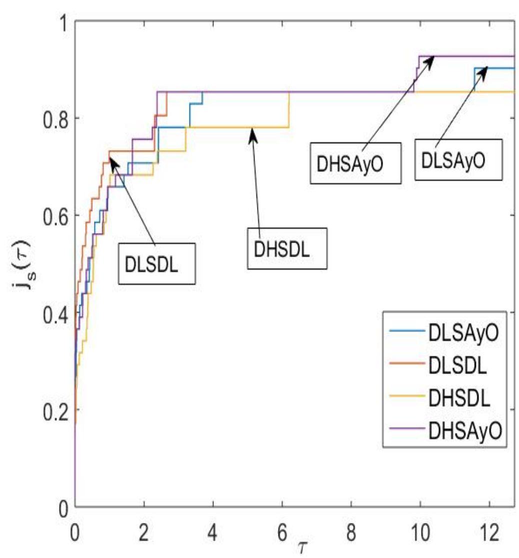

FIGURE 4. Gradient norm (G)

4.2. **Discussion of Results.** Reports of the experiments carried out on our proposed methods and other modified Dai-Liao methods are given in this section. In these experiments, we put into consideration the rate of convergence, the gradient norm, value of function and CPU time for the comparison.
All methods are tested on MATLAB R 2015a installed on a computer HP 650, windows 10 OS and RAM 3GB. α_k is computed in such a way that conditions (2.4)-(2.5) hold. We set $\delta = 0.0001$, $\sigma = 0.9$, $t = 0.1$ and $\mu = 1$ to run the algorithm codes for all the methods.
The test functions are taken from [2]. Tables 1, 2, 3 and 4 represent the numerical results of the test. The first column in the tables represents the name of the problem while IT, F, G and CPU stand for number of iterations, value of function, gradient norm and CPU time respectively. The dimensions used to test each problem are 500, 1000 and 10000. The results are reported in the form of IT/F and G/CPU. NAN indicates that the method concerned fails to solve the problem. Meanwhile, the stopping criterium is

$$\|g_k\| \leq 10^{-6}$$

for the iterations.
Zheng Yutao and Zheng Bing [14] compared DLSDL and DHSDL with improved Dai-Liao type methods, namely WYLDL [13], MHSDL [12] and MLSDL [8] methods and reported that DLSDL and DHSDL methods performed better than these three methods in term of CPU time, IT and function-gradient evaluation. As a result of this, we report in this study the behaviours of our proposed methods (DLSAyO and DHSAyO methods) against the DLSDL and DHSDL methods based on number of iterations, value function, gradient norm and CPU time.
Graphical representations of the behaviours of all tested methods given in TABLES 1, 2, 3 and 4 are presented in FIGURES 1-4. These figures represent the performance profiles of the four methods in term of IT, F, G and CPU time. These performance profiles are based on the performance profile designed by Dolan and More [6]. This is a programme designed to compare CG methods used in unconstrained optimization problems for the set of methods M to solve a set S of problems. Given n_m problems, there exits n_s methods.
If $\gamma_{m,s}$ is the IT, F, G or CPU time required to evaluate problem m by method s, then the ratio given by

$$i_{m,s} = \frac{\gamma_{m,s}}{min\{\gamma_{m,s} : s \in M \ and \ m \in S\}} \tag{4.1}$$

is used to compare the methods.
The cumulative distribution function for $i_{m,s}$ is denoted by

$$j_s(\tau) = \frac{1}{n_m}|m \in S : log\gamma_{m,s}| \leq \tau \tag{4.2}$$

with $\tau \geq 0$.
$j_s(\tau)$ is the probability that $i_{m,s}$ is within a factor $\tau \geq 1$ in relation to the method s. Furthermore, when $\tau = 1$, the method has the probability that it will perform better than other methods. On the other hand, a method $s \in M$ fails to solve a problem when $i_k = i_{m,s}$ for some parameter i_k.

FIGURES 1 and 2 show that DLSAyO method performed better than other methods in term of CPU time and number of iterations and it is followed closely by DHSAyO method. Both DLSAyO and DHSAyO methods compete considerably well with DLSDL and DHSDL methods in term of value of function and gradient norm. This can be observed in FIGURES 3 and 4.

5. Conclusion

This paper presented two new CG methods. The CG coefficients were constructed in such a way that the decent directions for the proposed methods were guaranteed under the strong Wolfe line search method. It was also shown that the methods have global convergence. Numerical analysis and results through the experiments performed proved that the proposed methods did well in term of convergence rate, value of function, gradient norm and CPU time. For further research work, these methods would be compared with some established efficient hybrid methods and also the value of parameter t would be varied to determine associated changes.

References

1. Abubakar, A. B., Malik, M., Kumam, P., Mohammad, H., Sun, M., Ibrahim, A. H., and Kiri, A. I. *A Liu-Storey-type conjugate gradient method for unconstrained minimization problem with application in motion control,* Journal of King Saud University-Science, **34(4)**, (2022), 101923.
2. Andrei N., *Open Problems in Nonlinear Conjugate Gradient Algorithms for Unconstrained Optimization,* Bulletin of the Malaysian Mathematical Sciences Society. **34(2)**, (2011), 319-330.
3. Ayinde, S.A., Osinuga, I. A., Adio, A. K. and Adelodun, J. F. *A new Dai-Liao-type conjugate gradient method for unconstrained optimization problems,* IAENG Engineering Letters, **31(3)**, (2023), 1281-1290.
4. Dai, Y.H. and L.Z. Liao, *New conjugacy conditions and related nonlinear conjugate gradient methods,* Applied Math. Optim. **43**, (2001), 87-101.
5. Dai, Y.H. and Y. Yuan, *A nonlinear conjugate gradient method with a strong global convergence property,* SIAM J. Optim. **10**, (1999), 177-182.
6. Dolan, E.D. and J.J. More, *Benchmarking optimization software with performance profiles,* Math. Program **91** (2002), 201-213.
7. Lu, J., Yuan, G. and Wang, Z., *A modified Dai-Liao conjugate gradient for solving unconstraied optimization and image restoration problems,* journal of Applied Mathematics and computing, **68**, (2022), 681-703.
8. Cheng, Y., Mou, Q., Pan, X., Yao, S., *A sufficient descent conjugate gradient method and its global convergence,* Optim. Methods Softw. **31(3)**, (2016), 577-590 . .
9. Li, G., Tang, C., Wei, Z., *New conjugacy condition and related new conjugate gradient methods for unconstrained optimization,* J. Comput. Appl. Math. **202(2)**, (2007), 523–539 .
10. Stanimirovic P. S., Ivanov B and Mosic D., *A survey of gradient methods for solving nonlinear optimization,* Electronic research archive **28(4)** (2020), 1573-1624.
11. Yabe, H., Takano, M., *Global convergence properties of nonlinear conjugate gradient methods with modified secant condition,* Comput. Optim. Appl. **28(2)**, (2004), 203–225
12. Yao, S., Lu, X., Wei, Z., *A conjugate gradient method with global convergence for large scale unconstrained optimization problems,* J. Appl. Math. (2013), Article ID:730454
13. Yao, S., Qin, B., *A hybrid of DL and WYL nonlinear conjugate gradient methods,* Abstr. Appl. Anal. (2014), Article ID: 279891

14. Zheng, Y. and Zheng, B. *Two new Dai-Liao-type conjugate gradient methods for unconstrained optimization problems,* J. Optim. Theory Appl., **175**, (2017), 502–509.
15. Zheng, Y. *A family of Dai-Liao conjugate gradient methods with modified secant equation for unconstrained optimization,* RAIRO Operations Research, **55**, (2021), 3281-3291.
16. Zoutendijk, G., *Nonlinear programming computational methods in Integer and Nonlinear Programming,* J. Abadie, ed., North-Holland, Amsterdam, (1970), 37–86.

[1] Department of Basic Sciences , Babcock University, Ilishan-Remo, Ogun State, Nigeria.
Email address: ayindes@babcock.edu.ng

[2] Department of Basic Sciences , Babcock University, Ilishan-Remo, Ogun State, Nigeria.
Email address: adelodunj@babcock.edu.ng

[4]Department of Mathematics, Faculty of Natural and Applied Sciences, Nigerian Army University Biu, Nigeria
Email address: agboolasunday70@gmail.com

[2] Department of Mathematical Sciences, Olabisi Onabanjo University, Ago Iwoye, Nigeria
Email address: samuel.hassan@oouagoiwoye.edu.ng