New Parameters of the Conjugate Gradient Method to Solve Nonlinear Systems of Equations

Rana Z. Al-Kawaz1 *, Abbas Y. Al-Bayati2
1Department of Mathematics, College of Basic Education, University of Telafer, Mosul, Iraq
2University of Telafer, Mosul, Iraq

Abstract

The conjugated gradient methods can solve smooth functions with large-scale variables in the specified number of iterations for that they are highly important methods compared to concerning other iterative methods. In this paper, we propose two new conjugate gradient methods, namely the PMDL-1 and PMDL-2. However, for non-smooth functions, which are called conjugate gradient-free derivative methods depending on the projection technique. The two methods give great results compared to the basic PDL method. Moreover, we provide theorems that prove the global convergence between these two methods.

Keywords: Conjugate gradient; Smooth functions; Projection technique; Free derivative methods.

1. Introduction

Let $F$ be a non-linear mapping, continuous and monotone function which is defined by $F: \Omega \subset \mathbb{R}^n \to \mathbb{R}^n$, where $\Omega$ is a nonempty closed and convex set and $\mathbb{R}^n$ is the n-dimensional Euclidean space. We say that $F$ is a monotone function if for any $x, y \in \Omega$, we have

*Email: ranazidan@uotelafer.edu.iq
Several methods have been proposed to solve the \( F(x) - F(y) \) \( (x - y) \geq 0 \). \( (1) \)

In this paper, we will find the solution to the following non-linear equation:
\[ F(x) = 0, \quad x \in \Omega. \] \( (2) \)

The methods for solving monotonous, non-linear, and unconstrained problems in case \( \Omega = \mathbb{R}^n \) are divided into Newton's method and semi-Newton methods. Their variables are very popular as a result of the convergence of local lines in them to the second and local degrees [1]-[6]. Whereas, the monotonic, non-linear equations are widespread, these methods are not good because they require solving linear system of equations which means, the Jacobian matrix of \( F(x) \) or rounding them in each iteration is used [7], [8]. Therefore, researchers examined to use the methods of the conjugate gradient for solving (2). We use the conjugate gradient algorithms to solve non-linear problems with large dimensions. These methods are well suited to these problems because of their demand for low memory, in addition to strong global convergence as an important feature of these algorithms. The iterative method for solving (2) usually has the general form:
\[ x_{k+1} = x_k + s_k. \] \( (3) \)

Where \( s_k = \alpha_k d_k, \alpha_k \) is the step length obtained by a suitable line search and \( d_k \) is the search direction. Examples of the first conjugated gradient algorithms can be found in [9]-[12]. These formulas were later developed by several researchers. Dai and Liao [13] gave the following formula
\[ \beta_k = \frac{\| h_k \|^2}{\| h_k \|^2 - \| \rho_h \|^2}. \] \( (4) \)

It is flexible due to the use of positive and different values of \( \tau \) and to change the direction of the search to obtain sufficient descent and global convergence properties [14]-[16]. Solodov and Svaiter [17] combined the Newton method and projection strategy, they proposed Newton's non-convergent, comprehensive method of a system of monotone equations without assuming contrast. Wang et. al. [18] extended Solodov and Svaiter's work to solve the constrained convex monotonic equations. Later, Ma and Wang [19] proposed a modified projection method to solve a system of monotonic equations with convex constraints. Although the projection methods for the restricted convex routine equations are proposed in [18] and [19] that have a very good numerical performance, however, they are not suitable for solving extensive monotonic equations because they require matrix storage. Recently, Liu and Li proposed a gradient multivariate synchronous spectrum projection algorithm for constraints in nonlinear monotony equations by combining a multivariate spectral gradient method with the Dai and Yuan (DY) conjugated gradient method. In addition, several methods have been proposed to solve the system of nonlinear monotone equations; for more details, see [20]-[26].

This paper is divided into the following sections: Section 2 gives the two suggested methods and their algorithms. Section 3: Some assumptions and conditions are applied to obtain global convergence. Finally, in Section 4, we present the numerical results of these algorithms.

2. Two new Parameters

In this section, we introduce two new updates to the modified Dai-Liao method (4) based on several changes that are shown in the following steps:

a) Modify and damped the parameter values \( \tau \) and \( y \). Consider this model for the following quadratic function:
\[ q_k(\tau) = f_{k-1} + \tau g_k^T d_{k-1} + \frac{\tau^2}{2} d_{k-1}^T \nabla^2 f_{k-1} d_{k-1}. \]

Since \( \epsilon \) is a very positive and small quantity, then the second derivative of the square function becomes:
\[ \nabla^2 f_{k-1} d_{k-1} \approx \frac{y_{k-1}}{\varepsilon} = \frac{g(x_{k-1} + \varepsilon d_{k-1}) - g(x_{k-1})}{\varepsilon} . \]

We replace
\[ y_{k-1} = \nabla f(x_k) - \nabla f(x_{k-1}) = g(x_{k-1} + \varepsilon d_{k-1}) - g(x_{k-1}) . \]

With damped techniques \([23-25]\) for \(y_k\) that is
\[ y_D^{k-1} = \omega_{k-1} y_{k-1} + (1 - \omega_{k-1}) B_{k-1} s_{k-1} , \]

so, we get:
\[ \nabla^2 f_{k-1} d_{k-1} \approx \frac{y_D^{k-1}}{\varepsilon} = \frac{1}{\varepsilon} (\omega_{k-1} y_{k-1} + (1 - \omega_{k-1}) B_{k-1} s_{k-1}) = \bar{y}_D^{k-1} . \]

If \(a = s_{k-1}^T y_{k-1} , b = s_{k-1}^T B_{k-1} s_{k-1}\) and
\[ \omega_{k-1} = \begin{cases} 0.8 b & \text{if } a \geq 0.2 b \\ b - a & \text{if } a < 0.2 b \end{cases} \]

The model becomes
\[ q_k(\tau) = f_{k-1} + \tau g_{k-1}^T d_{k-1} + \frac{\tau^2}{2} d_{k-1}^T \bar{y}_D^{k-1} , \]

which implies
\[ q_k' (\tau) = g_{k-1}^T d_{k-1} + \tau d_{k-1}^T \bar{y}_D^{k-1} , \quad 0 = g_{k-1}^T d_{k-1} + \tau d_{k-1}^T \bar{y}_D^{k-1} . \]

Hence,
\[ \tau_D^{k-1} = \frac{-g_{k-1}^T d_{k-1}}{d_{k-1}^T \bar{y}_D^{k-1}} = \frac{-g_{k-1}^T d_{k-1}}{s_{k-1}^T \bar{y}_D^{k-1}} , \]

and \(s_{k-1}^T \bar{y}_D^{k-1} = \frac{0.2}{\varepsilon} b\) implies
\[ \tau_D^{k-1} = \frac{\varepsilon |g_{k-1}^T d_{k-1}|}{0.2 s_{k-1}^T \bar{y}_D^{k-1}} . \]

If \(B_{k-1} = I\), then
\[ \tau_D^{k-1} = \frac{\varepsilon |g_{k-1}^T d_{k-1}|}{0.2 s_{k-1}^T s_{k-1}} . \]  \((5)\)

This \(\tau_D^{k-1}\) is very suitable for parameter Dai-Liao Eq. \((4)\).

b) Using the projection strategy for monotonic equations, the process needs to be accelerated using the monotonicity condition of \(F\). By monotonicity of \(F\) and letting \(z_k = x_k + \alpha_k d_k\), the hyperplane
\[ H_k = \{x \in \mathbb{R}^n | F(z_k)^T (x - z_k) = 0\} . \]  \((6)\)

Separates strictly \(x_k\) from the solution set of Eq. \((2)\). Through \([17]\) where the next iteration \(x_{k+1}\) is to be the projection of \(x_k\) onto the hyperplane \(H_k\). So, \(x_{k+1}\) can be evaluated as:
\[ x_{k+1} = P_{\Omega} \{ x_k - \xi_k F(z_k) \} = x_k - \frac{F(z_k)^T (x - z_k) F(z_k)}{||F(z_k)||^2} . \]  \((7)\)

Where \(\xi_k = \frac{F(z_k)^T (x - z_k)}{||F(z_k)||^2} . \)

Now, using the projection and damping technology, we can propose the new development of Dai-Liao as in the formula:
\[ \beta_k^{PMDL-1} = \frac{F_k^T (y_{k-1}^D - \tau_D^{k-1} s_{k-1})}{d_{k-1}^T s_{k-1}^D} , \]  \((8)\)

and \(\bar{y}_{k-1}^D = \frac{(\omega_{k-1} y_{k-1} + (1 - \omega_{k-1}) B_{k-1} s_{k-1})}{\xi_k} , \)
\[ \tau_D^{k-1} = \frac{\xi_k |F_k^T (x - z_k)|}{0.2 s_{k-1}^T s_{k-1}} . \]  \((9)\)

We have to note that it is possible to add an update to the denominator of the fraction in the first new formula Eq. \((8)\) and get good results as well, which we will present in the numerical results section of these two methods.
\[ \beta_k^{PMDL-2} = \frac{F_k^T (y_{k-1}^D - \tau_D^{k-1} s_{k-1})}{\delta ||F_k||^2 + (1 - \delta) ||d_{k-1}||^2} . \]  \((10)\)

So, the new search direction will be:

3935
\[ d_k = \begin{cases} -F(x_k) & \text{if } k = 0 \\ -F(x_k) + \beta^{PMDL-1}_k d_{k-1} & \text{if } k \geq 1 \end{cases} \]  

(11)

Where i=1,2. Now we can suggest the steps of the two new algorithms:

2.1 Algorithm (PMDL-1)

**Step 1:** Given \( x_0 \in \Omega \), \( \varphi \), \( \sigma \in (0,1) \), stop test \( \epsilon > 0 \), set \( k = 0 \).

**Step 2:** Evaluate \( F(x_k) \) and test if \( ||F(x_k)|| \leq \epsilon \) stop, else go to Step 3.

**Step 3:** Compute \( \tau^D_{k-1} \) and \( \rho^{PMDL-1}_k \) from Eq. (9), (8) respectively.

**Step 4:** Compute \( d_k \) by Eq. (11) and stop if \( d_k = 0 \).

**Step 5:** Set \( z_k = x_k + \alpha_k d_k \), where \( \alpha_k = \varphi^{m_k} \) with \( m_k \) being the smallest positive integer \( m_k \) such that

\[ F \left( x_k + \frac{\varphi^m}{\epsilon} d_k \right)^T d_k > -\sigma \frac{\varphi^m ||d_k||^2}{\epsilon ||F||^2} . \]  

(12)

**Step 6:** If \( z_k \in \Omega \) and \( ||F(z_k)|| \leq \epsilon \) stop, else compute the next point \( x_{k+1} \) from Eq. (7).

**Step 7:** Let \( k = k + 1 \) and go to Step 1.

2.2 Algorithm (PMDL-2)

**Step 1:** Given \( x_0 \in \Omega \), \( \varphi \), \( \delta \), \( \sigma \in (0,1) \), stop test \( \epsilon > 0 \), set \( k = 0 \).

**Step 2:** Evaluate \( F(x_k) \) and test if \( ||F(x_k)|| \leq \epsilon \) stop, else go to Step 3.

**Step 3:** Compute \( \tau^D_{k-1} \) and \( \rho^{PMDL-2}_k \) from Eq. (9) and (10), respectively.

**Step 4:** Compute \( d_k \) by (11) and stop if \( d_k = 0 \).

**Step 5:** Set \( z_k = x_k + \alpha_k d_k \), where \( \alpha_k = \varphi^{m_k} \) with \( m_k \) being the smallest positive integer \( m_k \) from Eq. (12).

**Step 6:** If \( z_k \in \Omega \) and \( ||F(z_k)|| \leq \epsilon \) stop, else compute the next point \( x_{k+1} \) from Eq. (7).

**Step 7:** Let \( k = k + 1 \) and go to Step 1.

3. Convergence Analysis

In this section, we establish the global convergence of the two new methods by using the following assumptions:

3.1 Assumptions

Suppose \( F \) fulfills the following assumptions:

(i) The solution group of the equation for Eq. (2) is non-empty.

(ii) The function \( F \) is Lipschitz continuous, i.e., there exists a positive constant \( L \) such that:

\[ ||F(x) - F(y)|| \leq L ||x - y||, \forall x, y \in \mathbb{R}^n \]  

(13)

(iii) \( F \) is uniformly monotone, that is,

\[ \langle F(x) - F(y), x - y \rangle \geq c ||x - y||^2, \forall x, y \in \mathbb{R}^n, c > 0 . \]  

(14)

3.2 Lemma

Using Assumptions 4.1 and \( d_k \) is known by Eq. (10), then the sufficient descent condition is held i.e.

\[ d_k^T F_k \leq -\rho_k ||F_k||^2 . \]  

(15)

Proof:

If \( k = 0 \) then \( d_k^T F_k = -F_k^T F_k = -||F_k||^2 \). For \( k > 0 \),

\[ d_k^T F_k = -||F_k||^2 + \frac{(F_k^T y^D_{k-1} d_{k-1})^T F_k}{d_{k-1}^T y^D_{k-1}} - \frac{(F_k^T y^D_{k-1})^T F_k}{d_{k-1}^T y^D_{k-1}}, \]

\[ d_k^T F_k = -||F_k||^2 + \frac{(F_k^T y^D_{k-1} d_{k-1}^T F_k)}{d_{k-1}^T y^D_{k-1}} - \frac{(F_k^T y^D_{k-1}) (d_{k-1}^T F_k)}{d_{k-1}^T y^D_{k-1}} \]
\[ d_k^T F_k = -\|F_k\|^2 + \frac{(F_k y_k^D - \frac{\zeta}{1 - \omega_k} d_{k-1}^T F_k)}{\|d_{k-1}^T y_k^D\|} (d_{k-1}^T F_k) \]

Since,

\[ F_k^T y_k^D = \frac{1}{\zeta_k} (\omega_k F_k^T y_{k-1} + (1 - \omega_k) F_k^T s_{k-1}) \]

\[ d_k^T F_k = -\|F_k\|^2 + \frac{\omega_k}{\zeta_k} d_{k-1}^T F_k + \frac{1}{\zeta_k} (1 - \omega_k) d_{k-1}^T d_{k-1}^T F_k \]

\[ d_k^T F_k = -\|F_k\|^2 + \frac{\omega_k}{\zeta_k} d_{k-1}^T F_k + \frac{1}{\zeta_k} (1 - \omega_k) \left( \frac{F_k^T s_{k-1}}{\|s_{k-1}\|^2} \right) (d_{k-1}^T F_k) \]

\[ d_k^T F_k = -\|F_k\|^2 + \frac{\omega_k}{\zeta_k} d_{k-1}^T F_k + \frac{1}{\zeta_k} (1 - \omega_k) \left( \frac{F_k^T s_{k-1}}{\|s_{k-1}\|^2} \right) \left( d_{k-1}^T F_k \right) \]

From the line search in Eq. (12) and the inequality

\[ -F_k^T (F_k - F_{k-1}) = F_k^T F_{k-1} - \|F_k\|^2 > -\|F_k\|^2 \]

So,

\[ d_k^T F_k > -\|F_k\|^2 + \frac{\omega_k}{\zeta_k} d_{k-1}^T F_k + \frac{1}{\zeta_k} (1 - \omega_k) \left( \frac{F_k^T s_{k-1}}{\|s_{k-1}\|^2} \right) \left( d_{k-1}^T F_k \right) \]

\[ d_k^T F_k > -\|F_k\|^2 + \frac{\omega_k}{\zeta_k} d_{k-1}^T F_k + \frac{1}{\zeta_k} (1 - \omega_k) \left( \frac{F_k^T s_{k-1}}{\|s_{k-1}\|^2} \right) \left( d_{k-1}^T F_k \right) \]

\[ i.e., \]

\[ \rho_k = 1 + \frac{\omega_k}{\zeta_k} d_{k-1}^T F_k + \frac{1}{\zeta_k} (1 - \omega_k) \left( \frac{F_k^T s_{k-1}}{\|s_{k-1}\|^2} \right) \left( d_{k-1}^T F_k \right) \]

This proves for the parameter-based algorithm \( \beta_k^{PMDL-1} \), we use the same steps for the second new algorithm \( \beta_k^{PMDL-2} \).

### 3.3 Lemma

Using Assumptions 4.1 and the sequence \( \{d_k\} \) is known by Eq. (10), then the bounded search direction is held i.e.

\[ \|F_k\| \leq \|d_k\| \leq \|F_k\| \]

Proof:

\[ \|d_k\| = -\|F_k\| + \beta_k^{PMDL-1} d_{k-1} \]

\[ \|d_k\| \leq \|F_k\| + \|F_k y_k^D - \frac{\zeta}{1 - \omega_k} d_{k-1}^T F_k^T s_{k-1} - \|d_{k-1}^T y_k^D\| \]

\[ \|d_k\| \leq \|F_k\| + \|F_k y_k^D - \frac{\zeta}{1 - \omega_k} d_{k-1}^T F_k^T s_{k-1}\| \]

\[ \|d_k\| \leq \|F_k\| + \|F_k y_k^D - \frac{\zeta}{1 - \omega_k} d_{k-1}^T F_k^T s_{k-1}\| \]

\[ \|d_k\| \leq 2\|\|F_k\| + \|F_k\| \frac{\|F_k^T s_{k-1}\|}{\|y_k^D - y_{k-1}\|} \]

By using \( |\tau_k^D| = \frac{|\zeta_k| |F_k^T s_{k-1}|}{0.2 \delta_k |s_{k-1}|} \) and \( |\zeta_k| > 0 \),

\[ \|y_k^D - y_{k-1}\| = \frac{1}{|\zeta_k|} (|\omega_k| |y_{k-1}| + (1 + |\omega_k| |y_{k-1}|)) \]

then the Eq. (17) turn to:
\[ \|d_k\| \leq \|F_k\| \left[ 2 + \frac{\zeta_k^2\|F_{k-1}\|}{0.2(\alpha_k|\omega_{k-1}|\|y_{k-1}\|+(1+|\omega_{k-1}|)\|s_{k-1}\|)} \right] \]

From Assumption 4.1 (ii) then:
\[ \|d_k\| \leq \|F_k\| \left[ 2 + \frac{\zeta_k^2\|F_{k-1}\|}{0.2\alpha_k\|s_{k-1}\|((1+L)|\omega_{k-1}|)} \right] \]

and \[|\omega_{k-1}| = \frac{0.8}{1+L}\]
\[\|d_k\| \leq \|F_k\| \left[ 2 + \frac{\zeta_k^2\|F_{k-1}\|}{0.2\alpha_k\|s_{k-1}\|((1.8))} \right] \]
\[\|d_k\| \leq \|F_k\| \left[ 2 + \frac{\zeta_k^2\|F_{k-1}\|}{0.36\|d_{k-1}\|} \right] \]

From the Dai-Liao search direction we can put \((d_{k-1} = -F_{k-1})\):
\[\|d_k\| \leq C\|F_k\|, \text{ i.e. } C = 2 + \frac{\zeta_k^2}{0.36}\]

The last result, which gives a sequence of search direction, is restricted by \(C\). This proves for the parameter-based algorithm \(\beta_k^{PM\text{DL}}\), we use the same steps for the second new algorithm \(\beta_k^{PM\text{DL}-2}\).

3.4 Lemma [17]
If \((\bar{x} \in \mathbb{R}^n)\) satisfy \(F(\bar{x}) = 0\) and \(\{x\}\) is generated by Algorithm 1 and 2 that check Lemma 4.3, then
\[\|x_{k+1} - \bar{x}\|^2 \leq \|x_k - \bar{x}\|^2 - \|x_{k+1} - x_k\|^2.\]

Specifically, \(\{x\}\) is bounded and
\[\sum_{k=0}^{\infty}\|x_{k+1} - x_k\| < \infty . \quad (18)\]

3.5 Lemma
Suppose \(\{x\}\) is generated by Algorithms 1 and 2, then
\[\lim_{k \to \infty} \alpha_k\|d_k\| = 0 \quad (19)\]

Proof:
From Lemma 4.4, which leads to the sequence \(\{\|x_k - \bar{x}\|\}\) which does not increase, convergent, and thus constrained. As well, \(\{x_k\}\) is bounded and \(\lim_{k \to \infty} \|x_{k+1} - x_k\| = 0\). From Eq. (15) and the line search Eq. (12), we have:
\[\|x_{k+1} - x_k\| = \left\| \frac{F(x_k)^T(x-x_k)}{\|F(x_k)\|^2} \|F(z_k)\| \right\| = \frac{\|d_k\|}{\|F(z_k)\|} \geq \alpha_k\|d_k\| \geq 0\]

Finally, we get:
\[\lim_{k \to \infty} \alpha_k\|d_k\| = 0.\]

Now we use all the previous Lemmas to demonstrate the global convergence of the two new algorithms.

3.6 Theorem
Let \(\{x_k\}\) and \(\{z_k\}\) be the sequences that are generated by Algorithms 1 and 2, then
\[\lim_{k \to \infty} \inf \|F(x_k)\| = 0 \quad (20)\]

Proof:
The proof will be divided into two cases:

**Case I:** If \(\liminf_{k \to \infty} \|d_k\| = 0\), we have \(\liminf_{k \to \infty} \|F(x_k)\| = 0.\)
If one uses the continuity of $F$, then the sequence $\{x_k\}$ has some accumulation point $\bar{x}$ such that $F(\bar{x}) = 0$. Since $\{\|x_k - \bar{x}\|\}$ converges and $\bar{x}$ is an accumulation point of $\{x_k\}$, it follows that it converges to $\bar{x}$.

**Case II:** If $\liminf_{k \to \infty} \|d_k\| > 0$, we have $\liminf_{k \to \infty} \|F(x_k)\| > 0$ and by (19), it holds that $\lim_{k \to \infty} \alpha_k = 0$. Using the line search

$$-F \left( x_k + \frac{\alpha}{\xi} d_k \right)^T d_k < \sigma \frac{\alpha}{\xi} \|d_k\|^2$$

and the boundedness of $\{x_k\},\{d_k\}$, we can choose a subsequence such that allowing $k$ to go infinity in the above inequality results

$$-F(\bar{x})^T \bar{d} \leq 0 .$$

(21)

On the other hand, from Eq. (15), we get

$$-F(\bar{x})^T \bar{d} \geq \rho_k \|F(\bar{x})\|^2 > 0 .$$

(22)

However, Eq. (21) and (22) imply a contradiction. So, it is $\liminf_{k \to \infty} \|F(x_k)\| > 0$ does not hold and the proof is complete.

4. Numerical Tests

In this section, we present the results of the implementation of the two new algorithms (PMDL-1 & PMDL-2), respectively, and compare the resulting values with the implementation of the same projection technique with the basic conventional Dai-Liao algorithm that is given by Eq. (4). All codes were implemented in the MATLAB R2018b program and were managed on the laptop with an intel COREi5 processor with 4GB of RAM and a CPU of 2.5GHZ. Program data for each algorithm $\varphi = 0.9, \delta = 0.1, \tau(PDL) = 0.26, \sigma = 0.2$. The results were compared by applying the 3 initial points i.e.:

$x_1 = (1,1,1,\ldots,1)^T , x_2 = \left( \frac{1}{2}, \frac{1}{2}, \ldots, \frac{1}{2} \right)^T , x_3 = \left( \frac{1}{10}, \frac{1}{10}, \ldots, \frac{1}{10} \right)^T$

that was implemented on 8 problems and tried these algorithms for several dimensions $n$ (1000, 5000, 7000, 12000). The stopping scale for the three algorithms is $\|F(x_k)\| < 10^{-8}$. Algorithms are distinguished by their performance in (Iter) the number of iterations, (Eval-F) the number of function evaluations, (Time) CPU time in seconds, and (Norm) the norm of the approximation solution. The test problems $F(x) = (f_1, f_2, f_3, \ldots, f_n)^T$ where $x = (x_1, x_2, x_3, \ldots, x_n)^T$, are listed as follows:

**Problem 1 [28]:** $F_i(x) = 2x_i - \sin|x_i|$, for $i = 1, 2, \ldots, n$ and $\Omega = \mathbb{R}^n_+.$

**Problem 2 [29]:** $F_i(x) = e^x_i - 1$, for $i = 1, 2, \ldots, n$ and $\Omega = \mathbb{R}^n_+.$

**Problem 3 [29]:** $F_i(x) = \ln(|x_i|) + 1 - \frac{x_i}{n}$, for $i = 1, 2, \ldots, n$ and $\Omega = \mathbb{R}^n.$

**Problem 4 [30]:** $F_i(x) = \min(\min(|x_i|, x_i^2), \max(|x_i|, x_i^2))$, for $i = 1, 2, \ldots, n$ and $\Omega = \mathbb{R}^n.$

**Problem 5 [31]:** $F_i(x) = e^{x_i - 1}, F_i(x) = e^{x_i - x_{i-1} - 1}$, for $i = 2, \ldots, n - 1$

**Problem 6 [32]:** $F_i(x) = \Sigma_{i=1}^n |x_i|$, for $i = 1, 2, \ldots, n$ and $\Omega = \mathbb{R}^n_+.$

**Problem 7 [32]:** $F_i(x) = \max |x_i|$, for $i = 1, 2, \ldots, n$ and $\Omega = \mathbb{R}^n_+.$

**Problem 8 [32]:** $F_i(x) = \Sigma_{i=1}^n |x_i| e^{-\Sigma_{i=1}^n \sin(x_i^2)}$, for $i = 1, 2, \ldots, n$ and $\Omega = \mathbb{R}^n_+.$
Figure 1: Performance results through several times calculation iterations

Figure 2: Performance results through several times calculation functions

Figure 3: Performance of results concerning the time taken to implement the algorithm
In this paper, we have implemented several (96) tests of these presented problems. We noticed that the performance of the two algorithms has alternated in terms of these characteristics:
1- As for the initial points, the third point is considered the best in most functions.
2- About dimensions, we notice that the second initial point is better for more n.
In general, the two new algorithms were the best for the points and dimensions that are given in this paper. Figures 1-3 show the performance of the first starting point in terms of dimensions and functions, and they also give a better performance.

5. Conclusions
In this paper, we have been able to demonstrate that the two new algorithms (Modified for Dai-Liao) are better for the 8 problems that are used within the paper and with the starting points that are suggested in the previous section. Through Figures 1 to 3, we can say that the two algorithms are the best according to the explained conditions for problems in achieving global convergence faster and with the least number of iterations compared to the basic algorithm and the projected method adopted within the paper. Also, theories used to prove the convergence of the two new methods gave greater efficiency to them.

6. Acknowledgements
The University of Mosul's Department of Mathematics in the College of Computer Science and Mathematics and the University of Telafer's Department of Mathematics and College of Basic Education are providing funding for this study. The authors state that they have no financial or personal stake in the outcome of this study.

References