A generalized Dai-Liao type CG-method with a new monotone line search for unconstrained optimization

Rana Z. Al-Kawaz[1], Abbas Y. Al-Bayati[2]
[1]Department of Mathematics, College of Basic Education, University of Telafer, Tall 'Afar, Mosul, Iraq
[2]College of Basic Education, University of Telafer, Tall 'Afar, Mosul, Iraq

Article Info

Article history:

Received Sep 07, 2022
Revised Oct 27, 2022
Accepted Nov 12, 2022

Keywords:

Dai-Liao CG-method
Global convergence property
Large-scale
Monotone line search
Optimum point

ABSTRACT

In this paper, we presented two developments, the first deals with generalizing the modernization of the damped Dai-Liao formula in an optimized manner. The second development is by suggesting a monotone line search formula for our new algorithms. These new algorithms with the new monotone line search yielded good numerical results when compared to the standard Dai-Liao (DL) and minimum description length (MDL) algorithms. Through several theorems, the new algorithms proved to have a faster convergence to reach the optimum point. These comparisons are drawn in the results section for the tools (Iter, Eval-F, Time) which are a comprehensive measure of the efficiency of the new algorithms with the basic algorithms that we used in the paper.

Corresponding Author:

Rana Z. Al-Kawaz
Department of Mathematics, College of Basic Education, University of Telafer
Tall 'Afar, Mosul, Iraq
Email: rana.alkawaz@yahoo.com

1. INTRODUCTION

Let us define the function (1):

$$F: \Omega \subset \mathbb{R}^n \to \mathbb{R}^n \tag{1}$$

Then the issue that we discuss in this article is (2):

$$F(x) = 0, x \in \Omega \tag{2}$$

When the solution vector $x \in \mathbb{R}^n$ and the function F is continuous and satisfy the monotonous inequality i.e.,

$$[F(x) - F(y)]^T(x - y) \geq 0 \tag{3}$$

There are several ways to solve this problem, such as the Newton method, and the procedure of conjugating gradients [1], [2]. Applications around this type of method have continued to this day [3]-[5]. Here, we talk about monotonous equations and their solving methods for their importance in many practical applications. For more details see [6]. The projection technique is one of the important methods used to find the solution to (1). The two researchers Solodov and Svaiter [7] gave their attention to the large-scale non-linear equations. Recently, many researchers implemented new articles on the topic of finding solutions to some constraints or unconstraint monotony equations. in various methods, so it had an aspect of interest as in [8]-[15]. The projection technique relies on being accelerated using a monotone case F and updating the new point using repetition:

$$z_k = x_k + \alpha_k d_k \tag{4}$$

As an initial iterative and the hyperplane is:

$$H_k = \{x \in \mathbb{R}^n | F(z_k)^T (x - z_k) = 0\} \tag{5}$$

To start using the projection technique, we use the update of the new point x_{k+1} as given in the [6] to be the projection of x_k onto the hyperplane H_k. So, can be evaluated (6).

$$x_{k+1} = P_\Omega[x_k - \xi_k F(z_k)] \text{ and } \xi_k = \frac{F(z_k)^T (x_k - z_k)}{\|F(z_k)\|^2} \tag{6}$$

Specifically, this document is laid out: specifically, we present the suggested generalized damped Dai-Liao (GDDL) method in section 2. A novel monotone line search was proposed, and the penalty of generalized Dai-Liao (DL) parameters was determined in section 3. Section 4 proves global convergence. In section 5 we provide the results of our numerical experiments.

2. GENERALIZED DAMPED DAI-LIAO

The thinking of many researchers was concerned with finding a suitable parameter for the method of the conjugate gradient (CG) and imposing conditions on it to make the search direction conjugate and reach the smallest point of the function faster and limited steps. One of these researchers was Dai-Liao who provided an appropriate parameter that always makes the direction of the search in a state accompanied by a parameter $\upsilon \in (0,1)$, for more details see [16].

$$\beta_k^{DL} = \frac{g_{k+1}^T y_k}{d_k^T y_k} - \upsilon \frac{g_{k+1}^T s_k}{d_k^T y_k} \tag{7}$$

Later Abubakar *et al.* [17] modified (6) by using the new formula of t and the projection technique in their formula. Fatemi [18] presented a generalized method for the Dai-Liao parameter and its derivation by applying the penalty function to the parameter to achieve a sufficient descent condition in the search direction. In this section we will rely on the same previous techniques with the addition of a damped quasi-Newton condition as in the following derivation:

$$q(d) = f_{k+1} + g_{k+1}^T d + \frac{1}{2} d^T B_{k+1} d \tag{8}$$

With $\nabla q(\alpha_{k+1} d_{k+1})$, the gradient of the model in x_{k+2}, as an estimation of g_{k+2}. It is easy to see that:

$$\nabla q(\alpha_{k+1} d_{k+1}) = g_{k+1} + \alpha_{k+1} B_{k+1} d \tag{9}$$

Unfortunately, α_{k+1} in (4) is not available in the current iteration, because d_{k+1} is unknown. Thus, we modified (4) and set.

$$g_{k+2} = g_{k+1} + t\, B_{k+1} d \tag{10}$$

Where $t > 0$ is a suitable approximation of α_{k+1} . If the search direction of the CG-method.

$$d_{k+1} = -g_{k+1} + \beta_k d_k \tag{11}$$

In an efficient nonlinear CG method, we introduce the following optimization problem based on the penalty function:

$$\min_{\beta_k} [g_{k+1}^T d_{k+1} + P \textstyle\sum_{i=1}^m [(g_{k+2}^T s_{k-i})^2 + (d_{k+1}^T y_{k-i})^2]] \tag{12}$$

If $m = 1, 2, 3, 4, 5, .., n$. Now, substituting (10), (11) in (12), and using the projection technique we obtain:

$$\min_{\beta_k} \Big[-\|F_{k+1}\|^2 + \beta_k g_{k+1}^T d_k + P \textstyle\sum_{i=1}^m [(F_{k+1}^T s_{k-i})^2 + 2t F_{k+1}^T s_{k-i}\, {d_{k+1}}^T B_{k+1} s_{k-i} + t^2 \left({d_{k+1}}^T B_{k+1} s_{k-i}\right)^2 + (F_{k+1}^T y_{k-i})^2 - 2\beta_k F_{k+1}^T y_{k-i} d_k^T y_{k-i} + (\beta_k d_k^T y_{k-i})^2] \Big] \tag{13}$$

After some algebraic abbreviations, we get the:

$$\beta_k = \frac{1}{\varphi}[-F_{k+1}^T d_k + 2P\sum_{i=1}^m F_{k+1}^T y_{k-i} d_k^T y_{k-i} + 2t^2 P\sum_{i=1}^m F_{k+1}^T B_{k+1} s_{k-i} d_k^T B_{k+1} s_{k-i} - 2tP\sum_{i=1}^m F_{k+1}^T s_{k-i}\, d_k^T B_{k+1} s_{k-i}] \tag{14}$$

Where:

$$\varphi = 2t^2 P\sum_{i=1}^m (d_k^T B_{k+1} s_{k-i})^2 + 2P\sum_{i=1}^m (d_k^T y_{k-i})^2$$

To get a new parameter, let us assume that the Hessian approximation B_{k+1} to satisfy the extended damped quasi-Newton (QN) equation and with the incorporation of the use of projection technology we get:

$$B_{k+1} s_{k-i} = \frac{1}{\xi_k}(\tau_k y_{k-i} + (1-\tau_k) B_k s_{k-i}) = \frac{1}{\xi_k} y_{k-i}^D = \bar{y}_{k-i}^D \tag{15}$$

And ξ_k is the projection step.

$$\tau_k = \begin{cases} 1 & if\ s_{k-i}^T y_{k-i} \geq s_{k-i}^T B_k s_{k-i} \\ \frac{\eta\, s_{k-i}^T B_k s_{k-i}}{s_{k-i}^T B_k s_{k-i} - s_{k-i}^T y_{k-i}} & if\ s_{k-i}^T y_{k-i} < s_{k-i}^T B_k s_{k-i} \end{cases} \tag{16}$$

Then,

$$\beta_k^{new} = \frac{-F_{k+1}^T d_k}{2P\sum_{i=1}^m \left(t^2 (d_k^T \bar{y}_{k-i}^D)^2 + (d_k^T y_{k-i})^2\right)} + \frac{\sum_{i=1}^m F_{k+1}^T y_{k-i} d_k^T y_{k-i}}{\sum_{i=1}^m \left(t^2 (d_k^T \bar{y}_{k-i}^D)^2 + (d_k^T y_{k-i})^2\right)} + \frac{t^2\sum_{i=1}^m F_{k+1}^T \bar{y}_{k-i}^D d_k^T \bar{y}_{k-i}^D}{\sum_{i=1}^m \left(t^2 (d_k^T \bar{y}_{k-i}^D)^2 + (d_k^T y_{k-i})^2\right)} - \frac{t\sum_{i=1}^m F_{k+1}^T s_{k-i} d_k^T \bar{y}_{k-i}^D}{\sum_{i=1}^m \left(t^2 (d_k^T \bar{y}_{k-i}^D)^2 + (d_k^T y_{k-i})^2\right)} \tag{17}$$

So, there are two possible scenarios for this parameter such that, case I: if $s_{k-i}^T y_{k-i} \geq s_{k-i}^T B_k s_{k-i}$ then $\tau_k = 1$ and $\bar{y}_{k-i}^D = \frac{y_{k-i}}{\xi_k}$.

$$\beta_k^{new1} = \frac{-F_{k+1}^T d_k}{2P\sum_{i=1}^m \left(\frac{t^2}{\xi_k^2}(d_k^T y_{k-i})^2 + (d_k^T y_{k-i})^2\right)} + \frac{\sum_{i=1}^m F_{k+1}^T y_{k-i} d_k^T y_{k-i}}{\sum_{i=1}^m \left(\frac{t^2}{\xi_k^2}(d_k^T y_{k-i})^2 + (d_k^T y_{k-i})^2\right)} + \frac{\frac{t^2}{\xi_k^2}\sum_{i=1}^m F_{k+1}^T y_{k-i} d_k^T y_{k-i}}{\sum_{i=1}^m \left(\frac{t^2}{\xi_k^2}(d_k^T y_{k-i})^2 + (d_k^T y_{k-i})^2\right)} - \frac{\frac{t}{\xi_k}\sum_{i=1}^m F_{k+1}^T s_{k-i} d_k^T y_{k-i}}{\sum_{i=1}^m \left(\frac{t^2}{\xi_k^2}(d_k^T y_{k-i})^2 + (d_k^T y_{k-i})^2\right)} \tag{18}$$

Put $s_k = \xi_k d_k$.

$$\beta_k^{new1} = \frac{\sum_{i=1}^m F_{k+1}^T y_{k-i} d_k^T y_{k-i}}{\sum_{i=1}^m (d_k^T y_{k-i})^2} - \frac{\xi_k t}{(t^2+1)} \frac{\sum_{i=1}^m F_{k+1}^T s_{k-i} d_k^T y_{k-i}}{\sum_{i=1}^m (d_k^T y_{k-i})^2} - \frac{\xi_k F_{k+1}^T s_k}{2P_1 (t^2+1)\sum_{i=1}^m (d_k^T y_{k-i})^2} \tag{19a}$$

To investigate the proposed method when P_1 approaches infinity, because by making this coefficient larger, we penalize the conjugacy condition and the orthogonality property violations more severely, thereby forcing the minimizer of (10) closer to that of the linear CG method.

$$\beta_k^{new1} = \frac{\sum_{i=1}^m F_{k+1}^T y_{k-i} d_k^T y_{k-i}}{\sum_{i=1}^m (d_k^T y_{k-i})^2} - \frac{\xi_k t}{(t^2+1)} \frac{\sum_{i=1}^m F_{k+1}^T s_{k-i} d_k^T y_{k-i}}{\sum_{i=1}^m (d_k^T y_{k-i})^2} \tag{19b}$$

When compared to (6), we notice that the value is:

$$\upsilon_k = \frac{\xi_k t}{(t^2+1)}, \upsilon \in (0, \frac{1}{2}) \tag{20}$$

Case II: if $s_{k-i}^T y_{k-i} < s_{k-i}^T B_k s_{k-i}$ then $\bar{y}_{k-i}^D = \frac{y_{k-i}^D}{\xi_k}$ from (12) and $s_k = \xi_k d_k$ by projection, the technique to convert the β_k^{new} form to:

$$\beta_k^{new2} = \left[\frac{t^2\sum_{i=1}^m F_{k+1}^T y_{k-i}^D d_k^T y_{k-i}^D}{\sum_{i=1}^m\left(t^2\left(d_k^T y_{k-i}^D\right)^2+\xi_k^2\left(d_k^T y_{k-i}\right)^2\right)} - \frac{t\xi_k \sum_{i=1}^m F_{k+1}^T s_{k-i} d_k^T y_{k-i}^D}{\sum_{i=1}^m\left(t^2\left(d_k^T y_{k-i}^D\right)^2+\xi_k^2\left(d_k^T y_{k-i}\right)^2\right)}\right] + \left[\frac{\sum_{i=1}^m F_{k+1}^T y_{k-i} d_k^T y_{k-i}}{\sum_{i=1}^m\left(t^2\left(d_k^T y_{k-i}^D\right)^2+\xi_k^2\left(d_k^T y_{k-i}\right)^2\right)} - \frac{\xi_k F_{k+1}^T s_k}{2P_2\sum_{i=1}^m\left(t^2\left(d_k^T y_{k-i}^D\right)^2+\xi_k^2\left(d_k^T y_{k-i}\right)^2\right)}\right] \quad (21)$$

Now using algebraic simplifications, we obtain:

$$\beta_k^{new2} = \frac{1}{\varphi^d}\left(\sum_{i=1}^m\left[t_1 F_{k+1}^T y_{k-i} - t_2 F_{k+1}^T s_{k-i}\right] + \sum_{i=1}^m \frac{d_k^T s_{k-i}}{d_k^T y_{k-i}} \sum_{i=1}^m\left[t_3 F_{k+1}^T y_{k-i} - t_4 F_{k+1}^T s_{k-i}\right] - \frac{\xi_k}{2P_2\sum_{i=1}^m d_k^T y_{k-i}} F_{k+1}^T s_k\right) \quad (22a)$$

i.e.,

$\varphi^d = \sum_{i=1}^m\left(t^2\left(d_k^T y_{k-i}^D\right)^2 + \xi_k^2\left(d_k^T y_{k-i}\right)^2\right)$

$t_1 = t^2\tau_k^2 + \xi_k^2$ and $t_2 = t\,\xi_k\tau_k - t^2\tau_k(1-\tau_k)$

$t_3 = t^2\tau_k(1-\tau_k)$ and $t_4 = t\,\xi_k(1-\tau_k) - t^2(1-\tau_k)^2)$

As we talked about (P_1) then (P_2) when you come close to infinity, then we use the parameter omitted from this limit:

$$\beta_k^{new2} = \frac{1}{\varphi^d}\left(\sum_{i=1}^m\left[t_1 F_{k+1}^T y_{k-i} - t_2 F_{k+1}^T s_{k-i}\right] + \sum_{i=1}^m \frac{d_k^T s_{k-i}}{d_k^T y_{k-i}} \sum_{i=1}^m\left[t_3 F_{k+1}^T y_{k-i} - t_4 F_{k+1}^T s_{k-i}\right]\right) \quad (22b)$$

To obtain better results as in section 5. We have another paper on this type of method, but without generalizing the original formula [17].

3. NEW PENALTY PARAMETER

Due to the importance of concomitant gradient methods in this paper, we highlight them in the derivation of new algorithms. Now, we will derive more coefficients of the penalty function. Although, in this formula, we will focus on the two new parameters defined in the (19) and (22) respectively, then we will check and update the derived parameters by relying on the appropriate direction of regression of the CG method:

3.1. Lemma 1

Assume that the sequence of the solution generated by the method (19) with monotone line search, then for a few positive scalars δ_1 and δ_2 satisfying $\delta_1 + \delta_2 < 1$, we have:

$$F_{k+1}^T d_{k+1} \le -(1-\delta_1-\delta_2)\|F_{k+1}\|^2 \quad (23)$$

When:

$$|\xi_k t - 1| \le \sqrt{\frac{2\,\delta_2\left(y_k^T s_k\right)}{\|s_k\|\sum_{i=1}^m\|s_{k-i}\|}} \quad (24)$$

$$P_1 = \frac{2\,\delta_1\|F_{k+1}\|^2}{m(t^2+1)\ \max\limits_{i=1,..,m}\left((y_{k-i}-0.5\,\lambda_i s_{k-i})^T F_{k+1}\right)^2}; \lambda_i \le 1 \quad (25)$$

Proof: if we used (5) and (14) then:

$$F_{k+1}^T d_{k+1} = -\|F_{k+1}\|^2 + \frac{1}{\sum_{i=1}^m\left(y_{k-i}^T s_k\right)^2}\left(\left(F_{k+1}^T y_k\right)\left(y_k^T s_k\right)\left(F_{k+1}^T s_k\right) - \frac{\xi_k t}{(t^2+1)}\left(F_{k+1}^T s_k\right)\left(y_k^T s_k\right)\left(F_{k+1}^T s_k\right)\right) - \frac{\xi_k}{2P_1(t^2+1)}\frac{\left(F_{k+1}^T s_k\right)^2}{\sum_{i=1}^m\left(y_{k-i}^T s_k\right)^2} \quad (26)$$

Since $\lambda_k \le 1$, implies that:

$$F_{k+1}^T d_{k+1} \le -\|F_{k+1}\|^2 + \frac{1}{\sum_{i=1}^m (y_{k-i}^T s_k)^2} \sum_{i=1}^m \Big(((y_{k-i} - 0.5\lambda_i s_{k-i})^T F_{k+1})(y_{k-i}^T s_k)(F_{k+1}^T s_k) \Big) + \left(\frac{1}{2} - \frac{\xi_k t}{(t^2+1)}\right) \frac{1}{\sum_{i=1}^m (y_{k-i}^T s_k)^2} \sum_{i=1}^m (F_{k+1}^T s_{k-i})(y_{k-i}^T s_k)(F_{k+1}^T s_k) - \frac{\xi_k}{2P_1(t^2+1)} \frac{(F_{k+1}^T s_k)^2}{\sum_{i=1}^m (y_{k-i}^T s_k)^2} \quad (27)$$

Using this inequality $xy \le \frac{t'}{4} x^2 + \frac{1}{t'} y^2$, where x, y and t' are positive scalars, we have:

$$F_{k+1}^T d_{k+1} \le -\|F_{k+1}\|^2 + \frac{t'}{4\sum_{i=1}^m (y_{k-i}^T s_k)^2} \sum_{i=1}^m ((y_{k-i} - 0.5\lambda_i s_{k-i})^T F_{k+1})^2 (y_{k-i}^T s_k)^2 + \frac{m}{\sum_{i=1}^m (y_{k-i}^T s_k)^2 t'} (F_{k+1}^T s_k)^2 - \frac{\xi_k}{2P_1 \sum_{i=1}^m (y_{k-i}^T s_k)^2 (t^2+1)} (F_{k+1}^T s_k)^2 + \frac{(\xi_k t - 1)^2}{2 y_k^T s_k (t^2+1)} \sum_{i=1}^m |s_{k-i}^T F_{k+1}|\, |F_{k+1}^T s_k| \quad (28)$$

Let $t' = 2mP_1(t^2+1)$,

$$F_{k+1}^T d_{k+1} \le -\|g_{k+1}\|^2 + \frac{P_1 m(t^2+1)}{2} \max_{i=1,\dots,m} ((y_{k-i} - 0.5\lambda_i s_{k-i})^T F_{k+1})^2 + \frac{(\xi_k t - 1)^2}{2 y_k^T s_k (t^2+1)} \sum_{i=1}^m |s_{k-i}^T F_{k+1}|\, |F_{k+1}^T s_k| \quad (29)$$

By Cauchy-Schwarz inequality implies:

$$F_{k+1}^T d_{k+1} \le -\left[1 - \frac{m P_1 (t^2+1)}{2\|F_{k+1}\|^2} \max_{i=1,\dots,m} ((y_{k-i} - 0.5\lambda_i s_{k-i})^T F_{k+1})^2 - \frac{(\xi_k t - 1)^2}{2 y_k^T s_k (t^2+1)} \|s_k\| \sum_{i=1}^m \|s_{k-i}\| \right] \|F_{k+1}\|^2 \quad (30)$$

$$F_{k+1}^T d_{k+1} \le -(1 - \delta_1 - \delta_2) \le -\rho_1 \|F_{k+1}\|^2 \quad (31)$$

Since t is an approximation of the step size, we use the updated formula:

$$t = \begin{cases} \xi_k \ \ if\ |\xi_k t - 1| \le \sqrt{\frac{2\,\delta_2 (y_k^T s_k)}{\|s_k\| \sum_{i=1}^m \|s_{k-i}\|}} \\ 1 + \sqrt{\frac{2\,\delta_2 (y_k^T s_k)}{\|s_k\| \sum_{i=1}^m \|s_{k-i}\|}} \qquad O.W. \end{cases} \quad (32)$$

Hence, the proof is completed.

3.2. Lemma 2

Assume that the solution sequence is generated by the new method (16) with a monotone line search, then for a few positive scalars $\delta_3, \delta_4, \delta_5$ and δ_6 satisfying $\delta_3 + \delta_4 + \delta_5 + \delta_6 < 1$, we have:

$$F_{k+1}^T d_{k+1} \le -(1 - \delta_3 - \delta_4 - \delta_5 - \delta_6)\|F_{k+1}\|^2 \quad (33)$$

$$|t_2 - 2| \le \sqrt{\frac{2\,\delta_4 \xi_k^2 (y_k^T s_k)}{\|s_k\| \sum_{i=1}^m \|s_{k-i}\|}} \ \ \&\ |t_4 - 2| \le \sqrt{2\,\delta_6 t^2 (1 - \tau_k)^2} \quad (34)$$

And

$$P_{2a} = \frac{2\,\delta_3 \|F_{k+1}\|^2}{m \max_{i=1,..,m} ((t_1\, y_{k-i} - 0.5\,\lambda_i s_{k-i})^T F_{k+1})^2} \text{ and } P_{2b} = \frac{2\,\delta_5 \|F_{k+1}\|^2}{m \max_{i=1,..,m} ((t_3\, y_{k-i} - 0.5\,\lambda_i s_{k-i})^T F_{k+1})^2} \quad (35)$$

$\lambda_i \le 1$ is a scalar.

Proof: we substituting (16) in (9) and multiplying by F_{k+1} that:

$$F_{k+1}^T d_{k+1} = -\|F_{k+1}\|^2 + \frac{1}{\varphi^d} \Big(\sum_{i=1}^m [t_1 (F_{k+1}^T y_{k-i})(y_{k-i}^T s_k)(F_{k+1}^T s_k) - t_2 (F_{k+1}^T s_{k-i})(y_{k-i}^T s_k)(F_{k+1}^T s_k)] + \sum_{i=1}^m \frac{d_k^T s_{k-i}}{d_k^T y_{k-i}} \sum_{i=1}^m [t_3 (F_{k+1}^T y_{k-i})(y_{k-i}^T s_k)(F_{k+1}^T s_k) - (F_{k+1}^T s_{k-i})(y_{k-i}^T s_k)(F_{k+1}^T s_k)] \Big) - \frac{\xi_k}{2P_2 \varphi^d} (F_{k+1}^T s_k)^2 \quad (36)$$

$$F_{k+1}^T d_{k+1} = -\|F_{k+1}\|^2 + \frac{1}{\varphi^d}\sum_{i=1}^m ((t_1 y_{k-i} - 0.5\lambda_i s_{k-i})^T F_{k+1})(y_{k-i}^T s_k)(F_{k+1}^T s_k) + \frac{1}{\varphi^d}\left[\frac{1}{2} - t_2\right]\sum_{i=1}^m (F_{k+1}^T s_{k-i})(y_{k-i}^T s_k)(F_{k+1}^T s_k) + \frac{1}{\varphi^d}\sum_{i=1}^m ((t_3 y_{k-i} - 0.5\lambda_i s_{k-i})^T F_{k+1})(s_k^T s_k)(F_{k+1}^T s_k) + \frac{1}{\varphi^d}\left[\frac{1}{2} - t_4\right]\sum_{i=1}^m (F_{k+1}^T s_{k-i})(s_k^T s_k)(F_{k+1}^T s_k) - \frac{\xi_k}{2P_2\varphi^d}(F_{k+1}^T s_k)^2 \quad (37)$$

By following the same steps in lemma 3.1 we get:

$$F_{k+1}^T d_{k+1} \le -\|F_{k+1}\|^2 + P_{2a}\sum_{i=1}^m ((t_1 y_{k-i} - 0.5\lambda_i s_{k-i})^T F_{k+1})^2 + \frac{(t_2-2)^2}{2\xi_k^2 y_k^T s_k}\sum_{i=1}^m |s_{k-i}^T F_{k+1}|\,|F_{k+1}^T s_k| + P_{2b}\sum_{i=1}^m ((t_3 y_{k-i} - 0.5\lambda_i s_{k-i})^T F_{k+1})^2 + \frac{(t_4-2)^2}{2t^2(1-\tau_k)^2 s_k^T s_k}\sum_{i=1}^m |s_{k-i}^T F_{k+1}|\,|F_{k+1}^T s_k| \quad (38)$$

By Cauchy-Schwarz inequality implies:

$$F_{k+1}^T d_{k+1} \le -\begin{bmatrix} 1 - mP_{2a}\max_{i=1,\dots,m}(t_1 y_{k-i} - 0.5\lambda_i s_{k-i})^2 - \frac{(t_2-2)^2}{2\xi_k^2 y_k^T s_k}\|s_k\|\sum_{i=1}^m\|s_{k-i}\| - \\ mP_{2b}\max_{i=1,\dots,m}(t_3 y_{k-i} - 0.5\lambda_i s_{k-i})^2 - \frac{(t_4-2)^2}{2t^2(1-\tau_k)^2 s_k^T s_k}\|s_k\|\sum_{i=1}^m\|s_{k-i}\| \end{bmatrix}\|F_{k+1}\|^2 \quad (39)$$

Then, $F_{k+1}^T d_{k+1} \le -(1-\delta_3-\delta_4-\delta_5-\delta_6)\|F_{k+1}\|^2 \le -\rho_2\|F_{k+1}\|^2$

The proof is completed. In the next paragraph we talk about the Algorithm 1 (minimum description length conjugate gradient (MDL-CG)) used for numerical comparison which is an update of the Dai-Liao algorithm:

Algorithm 1. MDL-CG [19]

Given $x_0 \in \Omega, r, \sigma \in (0,1)$, stop test $\epsilon > 0$, set $k = 0$.
Step 1: evaluate $F(x_k)$ and test if $\|F(x_k)\| \le \epsilon$ stop else goes to step 2.
Step 2: evaluate d_k by (9) and substitute $\beta_k^{MDL} = \frac{F_{k+1}^T y_k}{y_k^T s_k} - \nu_k\frac{F_{k+1}^T s_k}{y_k^T s_k}$ and $\nu_k = p\frac{\|y_k\|^2}{s_k^T y_k} - \mathrm{q}\frac{s_k^T y_k}{\|s_k\|^2}$. $d_k = 0$, is the stopping criterion.
Step 3: compute $z_k = x_k + \alpha_k d_k$, with the step-size $\alpha_k = r^{m_k}$.

$$-F(x_k + r^m d_k)^T d_k > \sigma r^m \|F(x_k + r^m d_k)\|\|d_k\|^2 \quad (40)$$

Step 4: check if $z_k \in \Omega$ and $\|F(z_k)\| \le \epsilon$ stop.
Step 5: let $k = k+1$ and go to step 1.

The new Algorithm 2 (GDDL-CG) is a generalization according to the value of m and it is an update and suppression of the Dai- Laio algorithm as in the steps:

Algorithm 2. GDDL-CG

Given $x_0 \in \Omega, \mathrm{r}, \sigma, \mu, \gamma \in (0,1)$, stop test $\epsilon > 0$, set $k = 0$.
Step 1: evaluate $F(x_k)$ and test if $\|F(x_k)\| \le \epsilon$ stop else goes to step 2.
Step 2: when $y_k^T s_k \ge s_k^T s_k$ compute P_1 from (25) and if $P_1 \ne \infty$ then β_k^{new1} from (19a) else (19b).
Step 3: when $y_k^T s_k < s_k^T s_k$ compute P_2 from (35) and if $P_2 \ne \infty$ then β_k^{new2} from (22a) else (22b).
Step 4: compute d_k by (9) and stop if $d_k = 0$.
Step 5: set $z_k = x_k + \alpha_k d_k$, where $\alpha_k = r^m$ with m to be the shortest positive number m so:

$$-F(x_k + \gamma r^{m_k} d_k)^T d_k > \sigma\gamma r^{m_k}\left[\mu\|d_k\|^2 + (1-\mu)\frac{\|d_k\|^2}{\|F_k\|^2}\right] \quad (41)$$

Step 6: if $z_k \in \Omega$ and $\|F(z_k)\| \le \epsilon$ stop, else compute the point x_{k+1} from (6).
Step 7: let $k = k+1$ and go to step 1.

4. GLOBAL CONVERGENCE

In the previous section, we gave a preface to the proof of convergence condition by establishing the property of sufficient descent through lemmas 3.1 and 3.2. In the beginning, there are a set of assumptions that we mention in this section, and then we move on to theories. Adding several lemmas for the step length of the new algorithm. Now we need some assumption, to begin with, the proof of convergence condition, which is illustrated thus:

4.1. Assumption

Suppose F fulfills the following assumptions:

a) The solution group of (2) is non-empty.

b) The function F is Lipschitz continuous, i.e.,

$$\|F(x) - F(y)\| \leq L\|x - y\|, \forall\, x, y \in \mathbb{R}^n \tag{42}$$

c) F satisfies,

$$\langle F(x) - F(y), x - y\rangle \geq c\|x - y\|^2, \forall\, x, y \in \mathbb{R}^n,\ c>0 \tag{43}$$

4.2. Lemma 1

Assume $(\bar{x} \in \mathbb{R}^n)$ satisfy $F(\bar{x}) = 0$ and $\{x\}$ is generated by the new algorithm GDDL-CG. If lemmas 3.1 and 3.2, hold, then $\|x_{k+1} - \bar{x}\|^2 \leq \|x_k - \bar{x}\|^2 - \|x_{k+1} - x_k\|^2$, and

$$\sum_{k=0}^{\infty} \|x_{k+1} - x_k\| < \infty \tag{44}$$

4.3. Lemma 2

Suppose $\{x\}$ is generated by the new algorithm GDDL-CG then:

$$\lim_{k\to\infty} \alpha_k \|d_k\| = 0 \tag{45}$$

Proof: since the sequence $\{\|x_k - \bar{x}\|\}$ is not increasing; $\{x_k\}$ is bounded, and $\lim_{k\to\infty} \|x_{k+1} - x_k\| = 0$. From (3) using a line search, we have:

$$\|x_{k+1} - x_k\| = \frac{|F(z_k)^T(x - z_k)|}{\|F(z_k)\|^2}\|F(z_k)\| = \frac{|\alpha_k F(z_k)^T d_k|}{\|F(z_k)\|} \geq \alpha_k\|d_k\| \geq 0 \tag{46}$$

Then the proof is completed.

4.4. Theorem

Let $\{x_k\}$ and $\{z_k\}$ be the sequences generated by the new algorithm GDDL-CG then:

$$\liminf_{k\to\infty} \|F(x_k)\| = 0 \tag{47}$$

Proof: case I: if $\liminf_{k\to\infty} \|d_k\| = 0$, then $\liminf_{k\to\infty} \|F(x_k)\| = 0$. The sequence $\{x_k\}$ has some accumulation point $\bar{x}$ such that $F(\bar{x}) = 0$. Hence, $\{\|x_k - \bar{x}\|\}$ converges to $\bar{x}$. Case II: if $\liminf_{k\to\infty} \|d_k\| > 0$, then $\liminf_{k\to\infty} \|F(x_k)\| > 0$. Hence $\lim_{k\to\infty} \alpha_k = 0$. Using the line search.

$$-F(x_k + \gamma r^{m_k} d_k)^T d_k > \sigma\gamma r^{m_k}\left[\mu\|d_k\|^2 + (1-\mu)\frac{\|d_k\|^2}{\|F_k\|^2}\right] \tag{48}$$

And the boundedness of $\{x_k\}, \{d_k\}$, yields

$$-F(\bar{x})^T \check{d} \leq 0 \tag{49}$$

From (17) and (23) we get:

$$-F(\bar{x})^T \check{d} \geq \rho_i \|F(\bar{x})\|^2 > 0 \tag{50}$$

The (31) and (32) indicates a contradiction for $i = 1, 2$. So, $\liminf_{k\to\infty} \|F(x_k)\| > 0$ does not hold and the proof is complete.

5. NUMERICAL PERFORMANCE

In this section, we present our numerical results that explain the importance of the new algorithm GDDL-CG compared to the MDL-CG algorithm [19] using Matlab R2018b program in a laptop calculator with its Core^{TM}i5 specifications. The program finds the results on several non-derivative functions through several two initial points. In Table 1, we review the details of the initial points used to compare algorithms as shown in Table 1.

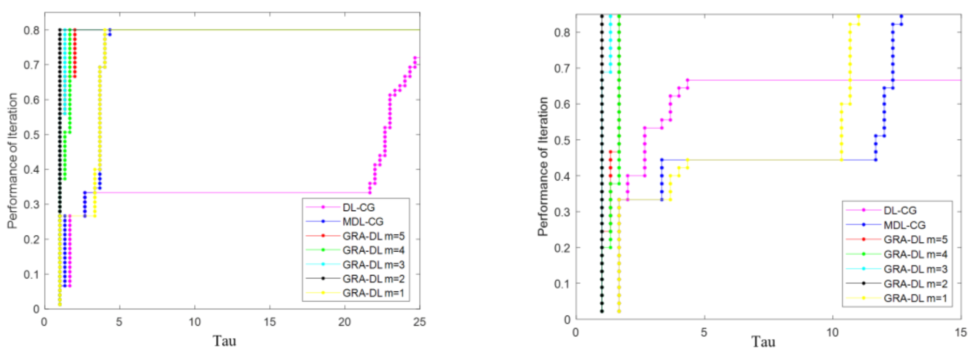
Table 1. Number of initial points

Name of variable	Number
x_1	$(1,1,1,..,1)^T$
x_2	$(\text{rand}, \text{rand}, \text{rand}, .., \text{rand})^T$

Table 2. Information of test functions [20]-[26]

Name of functions	Details	Reference
F_1	$F_i(x) = 2\,x_i - \sin\lvert x_i\rvert$	[21]
F_2	$F_i(x) = \ x_i - \sin(x_i)$	[21]
F_3	$F_i(x) = \ e^{x_i} - 1$	[22]
F_4	$F_i(x) = \sqrt{c}(x_1 - 1),\, for\ i = 2,3,..,n-1$ $F_n(x) = \frac{1}{4n}\sum_{j=1}^{n} x_j^2 - \frac{1}{4}$ $for\ c = 1 * 10^{-5}$	[22]
F_5	$F_i(x) = \ln(\lvert x_i\rvert + 1) - \frac{x_i}{n}$	[20]
F_6	$F_i(x) = \min(\min(\lvert x_i\rvert, x_i^2), \max(\lvert x_i\rvert, x_i^3))$	[23]
F_7	$F_1(x) = x_1 - e^{\frac{\cos(x_1+x_2)}{n+1}}$ $F_i(x) = x_i - e^{\frac{\cos(x_{i+1}+x_i+x_{i-1})}{n+1}},\, for\ i = 2,3,..,n-1$ $F_n(x) = x_n - e^{\frac{\cos(x_{n-1}+x_n)}{n+1}}$	[24]
F_8	$F_i(x) = \ \frac{i}{n} e^{x_i} - 1$	[21]
F_9	$F_1(x) = e^{x_1} - 1, F_i(x) = \ e^{x_i} - x_{i-1} - 1$	[20]
F_{10}	$F_i(x) = \sum_{i=1}^{n} \lvert x_i\rvert^i$	[25]
F_{11}	$F_i(x) = \sum_{i=1}^{n} \lvert x_i\rvert$	[25]
F_{12}	$F_i(x) = \max_{i=1,..,n} \lvert x_i\rvert$	[25]
F_{13}	$F_i(x) = \sum_{i=1}^{n} \lvert x_i\rvert\ e^{-\sum_{i=1}^{n}\sin(x_i^2)}$	[25]
F_{14}	$F_i(x) = \sum_{i=1}^{n} \lvert x_i \sin(x_i) + 0.1\,(x_i)\rvert$	[26]
F_{15}	$F_i(x) = \sum_{i=1}^{n} \lvert x_i\rvert^{i+1}$	[26]

The n-dimensional versions of these techniques are implemented here (1000, 2000, 5000, 7000, 12000). The stopping criterion is $\|F(x_k)\| < 10^{-8}$. All algorithms of this kind may be distinguished from one another based on their performance in terms of (Iter): the number of iterations, (Eval-F): the number of evaluations of functions, (Time): in seconds measured by the CPU, and (Norm): the approximate solution norm. In Table 2, we mention the details of the test problems $F(x) = (f_1, f_2, f_3, \dots, f_n)^T$ used with the references from which they were taken, the points $x = (x_1, x_2, x_3, \dots, x_n)^T$, and $\Omega = \mathbb{R}_+^n$.

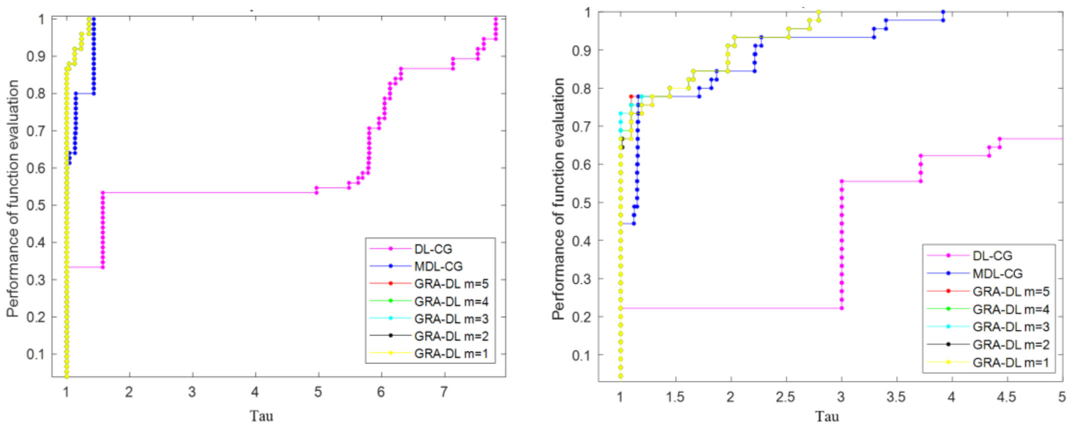


(a) (b)

Figure 1. Performance of the seven algorithms with respect to (Iter): (a) about x_1 and (b) about x_2

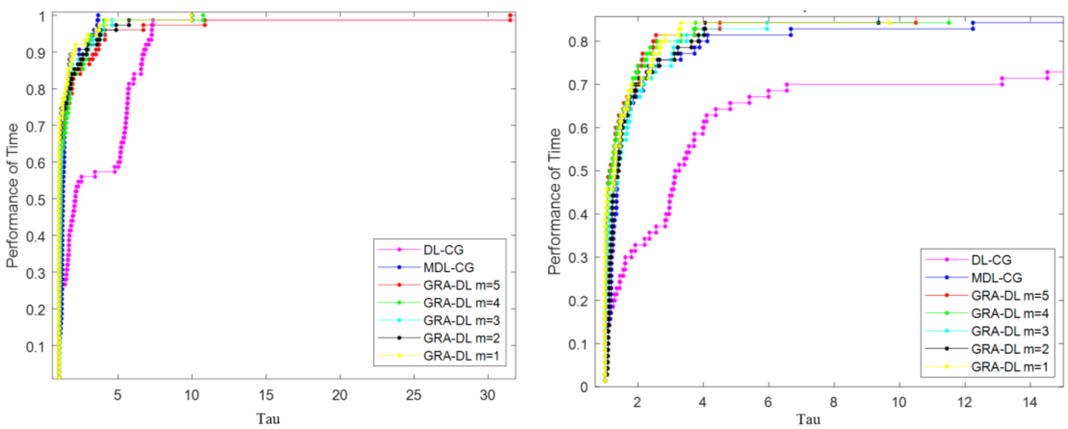
Using style for figures as in [19], the following three figures are for comparison between the old DL and MDL algorithms with the new algorithm when switching the value of the generalization i.e., from 1 to 5 to get 5 new algorithms. New algorithms are considered generalizations and can generalize to more than 5. To accurately know the impact of algorithms, we drew the following figures.

Figure 1 shows the effect of the new algorithms when taking the iterations tool as a measure of the effect, and the figure was divided into two parts, i.e., Figure 1(a) when calculating the point x_1, and Figure 1(b) when calculating the point x_2. As for Figure 2, it shows the effect of the new algorithms when taking the function number calculation tool as a measure of the effect, and the figure was divided into two parts, i.e., Figure 2(a) when calculating the point x_1, and Figure 2(b) when calculating the point x_2. Finally, Figure 3 shows the effect of the new algorithms when taking the time spent in calculations tool as a measure of impact, i.e., Figure 3(a) when calculating the point x_1, and Figure 3(b) when calculating the point x_2.



(a) (b)

Figure 2. Performance of the seven algorithms with respect to (Eval-F): (a) about x_1 and (b) about x_2



(a) (b)

Figure 3. Performing the seven algorithms with respect to (Time): (a) about x_1 and (b) about x_2

Through the use of the three figures, we can conclude that the new algorithms are superior to the two algorithms with which we compared our work. Furthermore, we have discovered that the random starting point with which we began our work brought to light the significance of the new algorithm when generalizing $m = 2$ when calculating the number of iterations and was the best of the new algorithms when calculating the number of times, the function is calculated. The algorithm with the value of m equal to one performed the best and was superior to the others. In conclusion, when considering the amount of effort invested in developing these algorithms, the two algorithms with m equal to 5 performed the best.

6. CONCLUSION

Our numerical results in the previous section, represented by three numbers, show the efficiency of the generalized algorithm GDDL when compared with the previous two algorithms. Its efficiency is better when we take the randomly generated point as a starting point and this will increase the efficiency when the dimensions used in the variables increase, which shows greater stability and makes the new generalized algorithm more suitable than other previous algorithms for the existence of the limit containing the penalty parameter in the new generalized algorithm. Theoretical proofs of the new algorithm give the proposed algorithms more power than the previous algorithms. This is why these algorithms are considered successful in both theoretical and numerical aspects.

ACKNOWLEDGEMENTS

The University of Mosul's Department of Computer Science and Mathematics and the University of Telafer's Department of Elementary and Secondary 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

[1] J. E. Dennis Jr and R. B. Schnabel, "Numerical methods for unconstrained optimization and nonlinear equations," *Classics in Applied Mathematics*, vol. 16, 1996, doi: 10.1137/1.9781611971200.

[2] J. Barzilai and J. M. Borwein, "Two-point step size gradient methods," *IMA Journal of Numerical Analysis*, vol. 8, pp. 141-148, 1988. [Online]. Available: https://pages.cs.wisc.edu/~swright/726/handouts/barzilai-borwein.pdf

[3] Y. Narushima, H. Yabe, and J. A. Ford, "A three-term conjugate gradient method with sufficient descent property for unconstrained optimization," *SIAM Journal on Optimization*, vol. 21, no. 1, pp. 212-230, 2011, doi: 10.1137/080743573.

[4] E. T. Hamed, R. Z. Al-Kawaz, and A. Y. Al-Bayati, "New Investigation for the Liu-Story Scaled Conjugate Gradient Method for Nonlinear Optimization," *Journal of Mathematics*, vol. 2020, 2020, doi: 10.1155/2020/3615208.

[5] A. B. Abubakar, P. Kumam, M. Malik, P. Chaipunya, and A. H. Ibrahim, "A hybrid FR-DY conjugate gradient algorithm for unconstrained optimization with application in portfolio selection," *AIMS Mathematics*, vol. 6, no. 6, pp. 6506-6527, 2021, doi: 10.3934/math.2021383.

[6] J. M. Ortega and W. C. Rheinboldt, "Iterative solution of nonlinear equations in several variables," *SIAM*, vol. 30, 2020, doi: 10.1137/1.9780898719468.

[7] M. V. Solodov and B. F. Svaiter, "A globally convergent inexact Newton method for systems of monotone equations," In *Reformulation: Nonsmooth, Piecewise Smooth, Semismooth, and Smoothing Methods*, Boston, MA: Springer, 1998, vol. 22, pp. 355-369, doi: 10.1007/978-1-4757-6388-1_18.

[8] W. Zhou and D. Shen, "An inexact PRP conjugate gradient method for symmetric nonlinear equations," *Numerical Functional Analysis and Optimization*, vol. 35, no. 3, pp. 370-388, 2014, doi: 10.1080/01630563.2013.871290.

[9] J. Liu and S. Li, "Spectral DY-type projection method for nonlinear monotone systems of equations," *Journal of Computational Mathematics*, vol. 33, pp. 341–355, 2015, doi: 10.4208/jcm.1412-m4494.

[10] S. -Y. Liu, Y. -Y. Huang, and H. -W. Jiao, "Sufficient descent conjugate gradient methods for solving convex constrained nonlinear monotone equations," *Abstract and Applied Analysis*, vol. 2014, 2014, doi: 10.1155/2014/305643.

[11] H. Mohammad and A. B. Abubakar, "A positive spectral gradient-like method for large-scale nonlinear monotone equations," *Bulletin of Computational Applied Mathematics*, vol. 5, no. 1, pp. 97–113, 2017. [Online]. Available: https://www.researchgate.net/publication/317932625_A_positive_spectral_gradient-like_method_for_large-scale_nonlinear_monotone_equations

[12] A. B. Abubakar, P. Kumam, H. Mohammad, A. M. Awwal, and K. Sitthithakerngkiet, "A modified Fletcher–Reeves conjugate gradient method for monotone nonlinear equations with some applications," *Mathematics*, vol. 7, no. 8, 2019, doi: 10.3390/math7080745.

[13] A. B. Abubakar and P. Kumam, "A descent Dai-Liao conjugate gradient method for nonlinear equations," *Numerical Algorithms*, vol. 81, pp. 197-210, 2019, doi: 10.1007/s11075-018-0541-z.

[14] R. Z. Al-Kawaz and A. Y. Al-Bayati, "An effective new iterative CG-method to solve unconstrained non-linear optimization issues," *TELKOMNIKA (Telecommunication, Computing, Electronics and Control)*, vol. 19, no. 6, pp. 1847-1856, 2021, doi: 10.12928/telkomnika.v19i6.19791.

[15] R. Z. Al-Kawaz and A. Y. Al-Bayati, "An adaptive search direction algorithm for the modified projection minimization optimization," *IOP Conference Series: Materials Science and Engineering*, 2021, vol. 1152, doi: 10.1088/1757-899X/1152/1/012019.

[16] Y. -H. Dai and L. -Z. Liao, "New conjugacy conditions and related nonlinear conjugate gradient methods," *Applied Mathematics and Optimization*, vol. 43, pp. 87-101, 2001, doi: 10.1007/s002450010019.

[17] A. B. Abubakar, P. Kumam, and A. M. Awwal, "A descent dai-Liao projection method for convex constrained nonlinear monotone equations with applications," *Thai Journal of Mathematics*, pp. 128-152, 2018. [Online]. Available: http://thaijmath.in.cmu.ac.th/index.php/thaijmath/article/view/3372

[18] M. Fatemi, "A new conjugate gradient method with an efficient memory structure," *Computational and Applied Mathematics*, vol. 38, pp. 59, 2019, doi: 10.1007/s40314-019-0834-4.

[19] W. La Cruz, J. M. Martínez, and M. Raydan, "Spectral residual method without gradient information for solving large-scale nonlinear systems of equations," *Mathematics of computation*, vol. 75, no. 255, pp. 1429-1448, 2006. [Online]. Available: https://www.ams.org/journals/mcom/2006-75-255/S0025-5718-06-01840-0/S0025-5718-06-01840-0.pdf

[20] W. Zhou and D. Li, "Limited memory BFGS method for nonlinear monotone equations," *Journal of Computational Mathematics*, vol. 25, no. 1, pp. 89-96, 2007. [Online]. Available: https://www.global-sci.org/v1/jcm/volumes/v25n1/pdf/251-89.pdf

[21] C. Wang, Y. Wang, and C. Xu, "A projection method for a system of nonlinear monotone equations with convex constraints," *Mathematical Methods of Operations Research*, vol. 66, pp. 33-46, 2007, doi: 10.1007/s00186-006-0140-y.

[22] R. Z. Al-Kawaz and A. Y. Al-Bayati, "Show off the efficiency of the dai-Liao method in merging technology for monotonous non-linear problems," *Indonesian Journal of Electrical Engineering and Computer Sciences*, vol. 21, no. 1, pp. 505-515, 2021, doi: 10.11591/ijeecs.v21.i1.pp505-515.
[23] W. La Cruz, "A spectral algorithm for large-scale systems of nonlinear monotone equations," *Numerical Algorithms*, vol. 76, pp. 1109-1130, 2017, doi: 10.1007/s11075-017-0299-8.
[24] X. -S. Yang, "Test Problems in Optimization," a*rXiv*, 2010, doi: 10.48550/arXiv.1008.0549.
[25] M. Jamil and X. -S. Yang, "A literature survey of benchmark functions for global optimization problems," *International Journal of Mathematical Modelling and Numerical Optimisation*, vol. 4, no. 2, pp. 150-194, 2013, doi: 10.48550/arXiv.1308.4008.
[26] E. D. Dolan and J. J. Mor´e, "Benchmarking optimization software with performance profiles," *Mathematical Programming*, vol. 91, pp. 201-213, 2002, doi: 10.1007/s101070100263.

BIOGRAPHIES OF AUTHORS

Rana Z. Al-Kawaz received the PhD. degree (Philosophy of Science) in computational numerical optimization from the Mosul University in Iraq. She is a Lecturer of Mathematical Sciences in the Department of Mathematics, College of Basic Education, University of Telafer, Mosul, Iraq. Additionally, she is working as the Head of Mathematics in College of Basic Education, University of Telafer, Mosul, Iraq. She has published over 16 journal papers, 1 authored book and 2 papers in conference proceedings. His research interests are in Numerical Optimization, Metahuristic Optimization Algorithm, Large-Scale Methods, Computational Numerical Analysis, Unconstrained Optimization, Conjugate Gradient Method, Nonlinear Equations, Operational Researches. She can be contacted at email: rana.alkawaz@yahoo.com and ranazidan@uotelafer.edu.iq.

Abbas Y. Al-Bayati received the PhD. degree (Philosophy of Science) in computational numerical optimization from the University of Leeds in U.K. He is a Professor of Mathematical Sciences in the Department of Mathematics, College of Basic Education, University of Telafer, Mosul, Iraq. In addition, he is currently the Director of the Computer Department at University of Telafer, Iraq. He has published over 235 journal papers, 1 authored book and 32 papers in conference proceedings. His research interests are in Numerical Analysis, Complex Analysis, Large-Scale Methods, Computational Numerical Optimization, Computer Oriented Algorithms, Parallel Computing, Operational Researches. He can be contacted at email: profabbasalbayati@yahoo.com.