Show off the efficiency of dai-liao method in merging technology for monotonous non-linear problems

Rana Z. Al-Kawaz, Abbas Y. Al-Bayati
Department of Mathematics, College of Basic Education, University of Telafer, Mosul, Iraq

Article Info

Article history:

Received May 16, 2020
Revised Jul 20, 2020
Accepted Aug 4, 2020

Keywords:

Dai-Liao method
Damping technology
Global convergence
Projection method
Quasi-Newton condition

ABSTRACT

In this article, we give a new modification for the Dai-Liao method to solve monotonous nonlinear problems. In our modification, we relied on two important procedures, one of them was the projection method and the second was the method of damping the quasi-Newton condition. The new approach of derivation yields two new parameters for the conjugated gradient direction which, through some conditions, we have demonstrated the sufficient descent property for them. Under some necessary conditions, the new approach achieved global convergence property. Numerical results show how efficient the new approach is when compared with basic similar classic methods.

Corresponding Author:

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

1. INTRODUCTION

The issue in this article is the assumption of finding the vector value x$\in \mathbb{R}^n$, i.e. as in:

$$x \in \Omega, \qquad F(x) = 0 \tag{1}$$

When $F:\Omega \subset \mathbb{R}^n \to \mathbb{R}^n$ is continuous and monotonous and satisfy $\left(F(x) - F(y)\right)^T (x - y) \geq 0$. Methods for solving this type of problem vary when they are not restricted to Newton's method and quasi-Newton methods, and they are preferred due to the convergence of their local lines to the second and local levels. When dealing with large-scale nonlinear equations, the so-called Conjugate Gradient (CG) method of all kinds is effective [1-8]. Applications and innovations continue around these technologies to this day [9-11]. The monotonic equations arose in several different practical situations for example see [12]. The most important advantage of CG-methods is that the direction of the search does not require the calculation of the Jacobin matrix which leads to low math requirements on each iteration. Likewise, when these methods overlap with the projection technique proposed by Solodov and Svaiter [13] to solve large-scale nonlinear equations and constrained nonlinear equations that some researchers have expanded as in [14-19]. Recently, many researchers have presented articles on how to find the solution to both constrained and unconstrained monotones (1) and give them a lot of attention [20-27]. Include the idea of projection that needs to be accelerated using a monotonous case F by monotony F and letting $z_k = x_k + \alpha_k d_k$, the hyperplane:

$$H = \{x \in \mathbb{R}^n | F(z_k)^T (x - z_k) = 0\} .$$

Separates strictly x_k from the solution set of (2). Through [13] where the next iteration x_{k+1} to be the projection of x_k onto the hyperplane H_k. So, x_{k+1} can be evaluated as:

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

$$\xi_k = \frac{F(z_k)^T(x_k-z_k)}{\|F(z_k)\|^2}$$

This paper is organized as follows: In Section 2, we describe the proposed new procedure. Section 3 derived the penalty parameters. Global convergence has been demonstrated in Section 4. For Section 5 we list numerical experiments in it.

2. OPTIMAL DAMPED DAI-LIAO

The researchers gave Dai and Liao [28] a parameter worthy of updating to this time because of its ability to reach global convergence through the properties of parameter t, which could set us some failures resulting from the deviation of the search direction of its path to reach the smallest point of the function, such that:

$$\beta_k^{DL} = \frac{g_{k+1}^T(y_k - t\, s_k)}{d_k^T y_k} \tag{3}$$

Later Liu and Li modified (3) by using the projection technique in their formula [29]. The researcher Fatemi [30] presented a precise method in deriving the conjugate gradient parameter β_k by setting conditions on it (the condition of orthogonality and conjugation) and through the penalty function the results of the derivation were largely appropriate in developing a formula for the Dai-Liao parameter and the positive value of t. In this section, we present an improved method for deriving a parameter β_k . That is:

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

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

$$\nabla q(\alpha_{k+1} d_{k+1}) = g_{k+1} + \alpha_{k+1} B_{k+1} d \tag{4}$$

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

$$g_{k+2} = g_{k+1} + t\, B_{k+1} d \tag{5}$$

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

$$d_{k+1} = -g_{k+1} + \beta_k d_k \tag{6}$$

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

$$\min_{\beta_k}[g_{k+1}^T d_{k+1} + P \textstyle\sum_{i=0}^{m} [(g_{k+2}^T s_{k-i})^2 + (d_{k+1}^T y_{k-i})^2]] \tag{7}$$

Now, substituting (5) and (6) in (7), and use the projection technique we obtain:

$$\min_{\beta_k}\Bigg[-\|F_{k+1}\|^2 + \beta_k g_{k+1}^T d_k + P\sum_{i=0}^{m}[(F_{k+1}^T s_{k-i})^2 + 2tF_{k+1}^T s_{k-i}\, {d_{k+1}}^T B_{k+1} s_{k-i} + t^2\left({d_{k+1}}^T B_{k+1} s_{k-i}\right)^2 + (F_{k+1}^T y_{k-i})^2 - 2\beta_k F_{k+1}^T y_{k-i} d_k^T y_{k-i} + (\beta_k d_k^T y_{k-i})^2]\Bigg]$$

After some algebraic abbreviations, we get the following formula

$$\beta_k = \frac{1}{\varphi}[-F_{k+1}^T d_k + 2P\sum_{i=0}^{m} F_{k+1}^T y_{k-i} d_k^T y_{k-i} + 2t^2 P \sum_{i=0}^{m} F_{k+1}^T B_{k+1} s_{k-i} d_k^T B_{k+1} s_{k-i} \\ -2tP\sum_{i=0}^{m} F_{k+1}^T s_{k-i}\, d_k^T B_{k+1} s_{k-i}] \quad (8)$$

Where $\varphi = 2t^2 P \sum_{i=0}^{m}(d_k^T B_{k+1} s_{k-i})^2 + 2P\sum_{i=0}^{m}(d_k^T y_{k-i})^2$

To get a new parameter we consider the following assumption as the hessian approximation B_{k+1} satisfies the extended damped quasi-Newton equation and with the incorporation of the use of projection technology we get:

$$B_{k+1} s_{k-i} = \frac{1}{\xi_k}(\tau_k y_{k-i} + (1-\tau_k) B_k s_{k-i}) = \frac{1}{\xi_k} y_{k-i}^D = \bar{y}_{k-i}^D \quad (9)$$

Where

$$\tau_k = \begin{cases} 1 & if\ s_{k-i}^T y_{k-i} \geq s_{k-i}^T B_k s_{k-i} \\ \frac{\eta\, s_{k-i}^T B_k s_{k-i}}{s_{k-i}^T B_k s_{k-i} - s_{k-i}^T y_{k-i}} & if\ s_{k-i}^T y_{k-i} < s_{k-i}^T B_k s_{k-i} \end{cases} \quad (10)$$

And ξ_k is the projection step. We get

$$\beta_k^{new} = \frac{-F_{k+1}^T d_k}{2P\sum_{i=0}^{m}(t^2(d_k^T \bar{y}_{k-i}^D)^2 + (d_k^T y_{k-i})^2)} + \frac{\sum_{i=0}^{m} F_{k+1}^T y_{k-i} d_k^T y_{k-i}}{\sum_{i=0}^{m}(t^2(d_k^T \bar{y}_{k-i}^D)^2 + (d_k^T y_{k-i})^2)} \\ + \frac{t^2\sum_{i=0}^{m} F_{k+1}^T \bar{y}_{k-i}^D d_k^T \bar{y}_{k-i}^D}{\sum_{i=0}^{m}(t^2(d_k^T \bar{y}_{k-i}^D)^2 + (d_k^T y_{k-i})^2)} - \frac{t\sum_{i=0}^{m} F_{k+1}^T s_{k-i}\, d_k^T \bar{y}_{k-i}^D}{\sum_{i=0}^{m}(t^2(d_k^T \bar{y}_{k-i}^D)^2 + (d_k^T y_{k-i})^2)}$$

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 $\bar{y}_{k-i}^D = \frac{y_{k-i}}{\xi_k}$

$$\beta_k^{new1} = \frac{-F_{k+1}^T d_k}{2P\sum_{i=0}^{m}\left(\frac{t^2}{\xi_k^2}(d_k^T y_{k-i})^2 + (d_k^T y_{k-i})^2\right)} + \frac{\sum_{i=0}^{m} F_{k+1}^T y_{k-i} d_k^T y_{k-i}}{\sum_{i=0}^{m}\left(\frac{t^2}{\xi_k^2}(d_k^T y_{k-i})^2 + (d_k^T y_{k-i})^2\right)} \\ + \frac{\frac{t^2}{\xi_k^2}\sum_{i=0}^{m} F_{k+1}^T y_{k-i} d_k^T y_{k-i}}{\sum_{i=0}^{m}\left(\frac{t^2}{\xi_k^2}(d_k^T y_{k-i})^2 + (d_k^T y_{k-i})^2\right)} - \frac{\frac{t}{\xi_k}\sum_{i=0}^{m} F_{k+1}^T s_{k-i}\, d_k^T y_{k-i}}{\sum_{i=0}^{m}\left(\frac{t^2}{\xi_k^2}(d_k^T y_{k-i})^2 + (d_k^T y_{k-i})^2\right)}$$

If $s_k = \xi_k d_k$

$$\beta_k^{new1} = \frac{\sum_{i=0}^{m} F_{k+1}^T y_{k-i} d_k^T y_{k-i}}{\sum_{i=0}^{m}(d_k^T y_{k-i})^2} - \frac{\xi_k t}{(t^2+1)}\frac{\sum_{i=0}^{m} F_{k+1}^T s_{k-i} d_k^T y_{k-i}}{\sum_{i=0}^{m}(d_k^T y_{k-i})^2} - \frac{\xi_k F_{k+1}^T s_k}{2P_1(t^2+1)\sum_{i=0}^{m}(d_k^T y_{k-i})^2} \quad (11)$$

It is interesting to investigate the 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 (7) closer to that of linear conjugate gradient method. We obtain

$$\beta_k^{new1} = \frac{\sum_{i=0}^{m} F_{k+1}^T y_{k-i} d_k^T y_{k-i}}{\sum_{i=0}^{m}(d_k^T y_{k-i})^2} - \frac{\xi_k t}{(t^2+1)}\frac{\sum_{i=0}^{m} F_{k+1}^T s_{k-i} d_k^T y_{k-i}}{\sum_{i=0}^{m}(d_k^T y_{k-i})^2} \quad (12)$$

We notice from the previous equation that it belongs to the parameter class for Dai-Liao and that's exactly when setting m=0 we have

$$\beta_k^{new1} = \frac{F_{k+1}^T y_k}{d_k^T y_k} - \frac{\xi_k t}{(t^2+1)} \frac{F_{k+1}^T s_k}{d_k^T y_k} - \frac{\xi_k F_{k+1}^T s_k}{2P_1(t^2+1) d_k^T y_k} \tag{13a}$$

$$\beta_k^{new1} = \frac{g_{k+1}^T y_k}{d_k^T y_k} - \frac{\xi_k t}{(t^2+1)} \frac{g_{k+1}^T s_k}{d_k^T y_k} \tag{13b}$$

When compared to the parameter Dai-Liao, we notice that the value is

$$\upsilon = \frac{\xi_k t}{(t^2+1)}, \quad \upsilon \in (0, \tfrac{1}{2}] \tag{14}$$

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 equation (9) and $s_k = \xi_k d_k$ by projection technique, m=0 to convert the β_k^{new} form to

$$\beta_k^{new2} = \left[\frac{\frac{t^2}{\xi_k^2} F_{k+1}^T y_k^D d_k^T y_k^D}{\left(\frac{t^2}{\xi_k^2}(d_k^T y_k^D)^2 + (d_k^T y_k)^2\right)} - \frac{\frac{t}{\xi_k} F_{k+1}^T s_k d_k^T y_k^D}{\left(\frac{t^2}{\xi_k^2}(d_k^T y_k^D)^2 + (d_k^T y_k)^2\right)} \right] + \left[\frac{F_{k+1}^T y_k d_k^T y_k}{\left(\frac{t^2}{\xi_k^2}(d_k^T y_k^D)^2 + (d_k^T y_k)^2\right)} - \frac{F_{k+1}^T s_k}{2P_2 \xi_k \left(\frac{t^2}{\xi_k^2}(d_k^T y_k^D)^2 + (d_k^T y_k)^2\right)} \right] \tag{15}$$

If we substituting equation (9) in (15) and using algebraic simplifications, we obtain the formula:

$$\beta_k^{new2} = \frac{1}{\varphi^d}\left([t_1 F_{k+1}^T y_k - t_2 F_{k+1}^T s_k] + \frac{d_k^T s_k}{d_k^T y_k}[t_3 F_{k+1}^T y_k - t_4 F_{k+1}^T s_k] - \frac{\xi_k}{2P_2 d_k^T y_k} F_{k+1}^T s_k \right) \tag{16a}$$

i.e. $\varphi^d = (t^2 (d_k^T y_k^D)^2 + \xi_k^2 (d_k^T y_k)^2)$

$t_1 = t^2 \tau_k^2 + \xi_k^2$ 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:

$$\beta_k^{new2} = \frac{1}{\varphi^d}\left([t_1 F_{k+1}^T y_k - t_2 F_{k+1}^T s_k] + \frac{d_k^T s_k}{d_k^T y_k}[t_3 F_{k+1}^T y_k - t_4 F_{k+1}^T s_k] \right) \tag{16b}$$

to obtain better results as in Section 5.

3. DERIVING THE PENALTY PARAMETER

The derivation will be according to the two new parameters defined in (13) and (16), which we will be updated by achieving the condition of a sufficient descent direction for the CG-method as shown below:

3.1. Lemma

Assume that the generated method (13) with line search, then for a few positive scalars δ_1 and δ_2 satisfying $\delta_1 + \delta_2 < 1$, we have:

$$F_{k+1}^T d_{k+1} \le -(1 - \delta_1 - \delta_2) \|F_{k+1}\|^2 \tag{17}$$

When

$$|\xi_k t - 1| \le \sqrt{\frac{2\, \delta_2 (y_k^T s_k)}{\|s_k\|^2}} \tag{18}$$

$$P_1 = \frac{2\, \delta_1 \|F_{k+1}\|^2}{(t^2+1)\, \|(y_k - 0.5\, \lambda_k s_k)^T F_{k+1}\|^2}; \quad \lambda_k \le 1 \text{ is a scalar.} \tag{19}$$

Proof:
We have used (2) and (13) that:

$$F_{k+1}^T d_{k+1} = -\|F_{k+1}\|^2 + \frac{1}{(y_k^T s_k)^2}\left((F_{k+1}^T y_k)(y_k^T s_k)(F_{k+1}^T s_k) - \frac{\xi_k t}{(t^2+1)}(F_{k+1}^T s_k)(y_k^T s_k)(F_{k+1}^T s_k)\right) - \frac{\xi_k}{2P_1(t^2+1)}\frac{(F_{k+1}^T s_k)^2}{(y_k^T s_k)^2} \quad (20)$$

since $\lambda_k \leq 1$, implies that

$$F_{k+1}^T d_{k+1} \leq -\|F_{k+1}\|^2 + \frac{1}{(y_k^T s_k)^2}\left(((y_k - 0.5\lambda_k s_k)^T F_{k+1})(y_k^T s_k)(F_{k+1}^T s_k)\right) - \frac{\xi_k}{2P_1(t^2+1)}\frac{(F_{k+1}^T s_k)^2}{(y_k^T s_k)^2} + \left(\frac{1}{2} - \frac{\xi_k t}{(t^2+1)}\right)\frac{1}{(y_k^T s_k)^2}(F_{k+1}^T s_k)(y_k^T s_k)(F_{k+1}^T s_k)$$

Now, using the following inequality:

$$xy \leq \frac{t'}{4}x^2 + \frac{1}{t'}y^2 \quad (21)$$

Where x, y and t' are positive scalars, we have:

$$F_{k+1}^T d_{k+1} \leq -\|F_{k+1}\|^2 + \frac{t'}{4(y_k^T s_k)^2}((y_k - 0.5\lambda_k s_k)^T F_{k+1})^2 (y_k^T s_k)^2 + \frac{1}{(y_k^T s_k)^2 t'}(F_{k+1}^T s_k)^2 - \frac{\xi_k}{2P_1(y_k^T s_k)^2(t^2+1)}(F_{k+1}^T s_k)^2 + \frac{(\xi_k t - 1)^2}{2y_k^T s_k(t^2+1)}(F_{k+1}^T s_k)^2$$

Let $t' = 2P_1(t^2+1)$,

$$F_{k+1}^T d_{k+1} \leq -\|g_{k+1}\|^2 + \frac{P_1(t^2+1)}{2}((y_k - 0.5\lambda_k s_k)^T F_{k+1})^2 + \frac{(\xi_k t - 1)^2}{2y_k^T s_k(t^2+1)}(F_{k+1}^T s_k)^2$$

By Cauchy-Schwarz inequality implies:

$$F_{k+1}^T d_{k+1} \leq -\left[1 - \frac{P_1(t^2+1)}{2\|F_{k+1}\|^2}((y_k - 0.5\lambda_k s_k)^T F_{k+1})^2 - \frac{(\xi_k t - 1)^2}{2y_k^T s_k(t^2+1)}\|s_k\|^2\right]\|F_{k+1}\|^2$$

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

i.e.

$$P_1 = \frac{2\,\delta_1\|F_{k+1}\|^2}{(t^2+1)\,\|(y_k - 0.5\,\lambda_k s_k)^T F_{k+1}\|^2} \qquad \text{when} \qquad |\xi_k t - 1| \leq \sqrt{\frac{2\,\delta_2(y_k^T s_k)}{\|s_k\|^2}}$$

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

$$t = \begin{cases} \xi_k & if \quad |\xi_k t - 1| \leq \sqrt{\frac{2\,\delta_2(y_k^T s_k)}{\|s_k\|^2}} \\ 1 + \sqrt{\frac{2\,\delta_2(y_k^T s_k)}{\|s_k\|^2}} & O.W. \end{cases} \quad (22)$$

Now the proof is completed.

3.2. Lemma

Assume that the newly generated method (16) with 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:

$$F_{k+1}^T d_{k+1} \leq -(1 - \delta_3 - \delta_4 - \delta_5 - \delta_6)\|F_{k+1}\|^2 \quad (23)$$

When

$$|t_2 - 2| \le \sqrt{\frac{2\,\delta_4 \xi_k^2 (y_k^T s_k)}{\|s_k\|^2}} \quad \& \quad |t_4 - 2| \le \sqrt{2\,\delta_6 t^2 (1-\tau_k)^2} \tag{24}$$

$$\text{and } P_{2a} = \frac{2\,\delta_3 \|F_{k+1}\|^2}{\|(t_1 y_k - 0.5\,\lambda_k s_k)^T F_{k+1}\|^2} \quad \& \quad P_{2b} = \frac{2\,\delta_5 \|F_{k+1}\|^2}{\|(t_3 y_k - 0.5\,\lambda_k s_k)^T F_{k+1}\|^2} \tag{25}$$

$\lambda_k \le 1$ is a scalar.

Proof:
We substituting (16) in (6) and multiplying by F_{k+1} that:

$$F_{k+1}^T d_{k+1} = -\|F_{k+1}\|^2 + \frac{1}{\varphi^d}\Big([t_1(F_{k+1}^T y_k)(y_k^T s_k)(F_{k+1}^T s_k) - t_2(F_{k+1}^T s_k)(y_k^T s_k)(F_{k+1}^T s_k)] + \frac{d_k^T s_k}{y_k^T s_k}[t_3(F y_k)(y_k^T s_k)(F_{k+1}^T s_k) - t_4(F_{k+1}^T s_k)(y_k^T s_k)(F_{k+1}^T s_k)]\Big) - \frac{\xi_k}{2P_2\varphi^d}(F_{k+1}^T s_k)^2$$

$$F_{k+1}^T d_{k+1} = -\|F_{k+1}\|^2 + \frac{1}{\varphi^d}((t_1 y_k - 0.5\lambda_k s_k)^T F_{k+1})(y_k^T s_k)(F_{k+1}^T s_k) + \frac{1}{\varphi^d}\left[\frac{1}{2} - t_2\right](F_{k+1}^T s_k)(y_k^T s_k)(F_{k+1}^T s_k) + \frac{1}{\varphi^d}((t_3 y_k - 0.5\lambda_k s_k)^T F_{k+1})(s_k^T s_k)(F_{k+1}^T s_k) + \frac{1}{\varphi^d}\left[\frac{1}{2} - t_4\right](F_{k+1}^T s_k)(s_k^T s_k)(F_{k+1}^T s_k) - \frac{\xi_k}{2P_2\varphi^d}(F_{k+1}^T s_k)^2$$

By following the same steps in Lemma 3.1 we get:

$$F_{k+1}^T d_{k+1} \le -\|F_{k+1}\|^2 + P_{2a}((t_1 y_k - 0.5\lambda_k s_k)^T F_{k+1})^2 + \frac{(t_2-2)^2}{2\xi_k^2 y_k^T s_k}(F_{k+1}^T s_k)^2 + P_{2b}((t_3 y_k - 0.5\lambda_k s_k)^T F_{k+1})^2 + \frac{(t_4-2)^2}{2t^2(1-\tau_k)^2 s_k^T s_k}(F_{k+1}^T s_k)^2$$

By Cauchy-Schwarz inequality implies:

$$F_{k+1}^T d_{k+1} \le -\left[1 - P_{2a}\|t_1 y_k - 0.5\lambda_k s_k\|^2 - \frac{(t_2-2)^2}{2\xi_k^2 y_k^T s_k}\|s_k\|^2 - P_{2b}\|t_3 y_k - 0.5\lambda_k s_k\|^2 - \frac{(t_4-2)^2}{2t^2(1-\tau_k)^2 s_k^T s_k}\|s_k\|^2\right]\|F_{k+1}\|^2$$

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.

3.3. Algorithm (PDL-CG) [29]

Given $x_0 \in \Omega, \gamma > 0, r,\ \sigma, \varepsilon \in (0,1)$, stop test $\epsilon > 0$, set $k = 0$.
Step1: Evaluate $F(x_k)$ and test if $\|F(x_k)\| \le \epsilon$ stop, else go to Step 3.
Step2: Generate the search direction d_k by (6) and here $\beta_k^{PDL} = \frac{F_{k+1}^T y_k}{w_k^T s_k} - 2\|y_k\|^2 \frac{F_{k+1}^T s_k}{w_k^T s_k}$; $w_k = y_k + t_k d_k$ and $t_k = 1 + \max\left(0, -\frac{y_k^T d_k}{d_k^T d_k}\right)$. Stop if $d_k = 0$.
Step3: Set $z_k = x_k + \alpha_k d_k$, where the step-size $\alpha_k = \max(\gamma r^m | m = 0,1,2,..)$ is determined by the line search $-F(x_k + \alpha_k d_k)^T d_k > \sigma \alpha_k \|d_k\|^2$
Step4: If $z_k \in \Omega$ and $\|F(z_k)\| \le \epsilon$ stop, else compute the next point x_{k+1} from Step (2).
Step5: Let $k = k + 1$ and go to Step 1.

3.4. New Algorithm (NDDL-CG).
Step1: Given $x_0 \in \Omega, r, \eta, \sigma, \varepsilon \in (0,1)$, stop test $\epsilon > 0$, set $k = 0$.
Step2: Evaluate $F(x_k)$ and test if $\|F(x_k)\| \leq \epsilon$ stop, else go to Step 3.
Step3: When $y_k^T s_k \geq s_k^T s_k$ compute P_1 from (19) and if $P_1 \neq \infty$ then β_k^{new1} from (13a) else from (13b).
Step4: When $y_k^T s_k < s_k^T s_k$ compute P_2 from (25) and if $P_2 \neq \infty$ then β_k^{new2} from (16a) else from (16b).
Step5: Compute d_k by (6) and stop if $d_k = 0$.
Step6: Set $z_k = x_k + \alpha_k d_k$, where $\alpha_k = r^m$ with m being the smallest positive integer m such that:

$$-F\left(x_k + \frac{r^m}{\varepsilon} d_k\right)^T d_k > \sigma \frac{r^m}{\varepsilon} \|d_k\|^2$$

Step7: If $z_k \in \Omega$ and $\|F(z_k)\| \leq \epsilon$ stop, else compute the next point x_{k+1} from (2).
Step8: Let $k = k + 1$ and go to Step1.

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. 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., there exists a positive constant L such that:

$$\|F(x) - F(y)\| \leq L\|x - y\|, \forall\, x, y \in \mathbb{R}^n \quad (26)$$

c) F is uniformly monotone, that is,

$$\langle F(x) - F(y), x - y\rangle \geq c\|x - y\|^2, \forall\, x, y \in \mathbb{R}^n, c>0 \quad (27)$$

4.2. Lemma [13]

assume $(\bar{x} \in \mathbb{R}^n)$ satisfy $F(\bar{x}) = 0$ and $\{x\}$ is generated by the new algorithm (NDDL-CG) that check **Lemmas** 3.1 and 3.2, then $\|x_{k+1} - \bar{x}\|^2 \leq \|x_k - \bar{x}\|^2 - \|x_{k+1} - x_k\|^2$. Specifically, it is $\{x\}$ bounded and

$$\sum_{k=0}^{\infty} \|x_{k+1} - x_k\| < \infty \quad (28)$$

4.3. Lemma

Suppose $\{x\}$ is generated by the new algorithm (NDDL-CG) then

$$\lim_{k\to\infty} \alpha_k \|d_k\| = 0 \quad (29)$$

<u>Proof:</u>

The sequence $\{\|x_k - \bar{x}\|\}$ not increasing, converging, and thus constrained. As well, $\{x_k\}$ is bounded and $\lim_{k\to\infty}\|x_{k+1} - x_k\| = 0$. From (2) and used a line search, we have:

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

Then the proof is completed.

4.4. Theorem.

Let $\{x_k\}$ and $\{z_k\}$ be the sequences generated by the new algorithm (NDDL-CG) then

$$\liminf_{k\to\infty} \|F(x_k)\| = 0. \quad (30)$$

<u>Proof:</u>
Case I: If $\liminf \|d_k\|_{k\to\infty} = 0$, we have $\liminf_{k\to\infty} \|F(x_k)\| = 0$. We use the continuity of F, 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 converges to $\bar{x}$.

Case II: If $\liminf \|d_k\|_{k\to\infty} > 0$, we have $\liminf_{k\to\infty} \|F(x_k)\| > 0$. By (29), it holds that $\lim_{k\to\infty} \alpha_k = 0$. Using the line search:

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

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

$$-F(\bar{x})^T \check{d} \leq 0 \quad (31)$$

On the other hand, from (17) and (23) we get

$$-F(\bar{x})^T \check{d} \geq \rho_i \|F(\bar{x})\|^2 > 0 \quad (32)$$

For i=1 and 2. It is through (31) and (32) indicates a contradiction. So, it is lim inf $_{k\to\infty} \|F(x_k)\| > 0$ does not hold and the proof is complete.

4. RESULTS AND EXPLANATIONS

In this section, we present several results that explain the importance of the new algorithm (NDDL-CG) compared to the standard Dai-Liao (PDL-CG) algorithm [20] 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 primary points indicated asshown in Table 1.

Table 1. Number of initial points

Name of Variable	Number
x_1	$(1,1,1,..,1)^T$
x_2	$(10,10,10,..,10)^T$
x_3	$(\frac{1}{2},\frac{1}{2},\frac{1}{2},\ldots,\frac{1}{2})^T$
x_4	$(\frac{1}{n},\frac{1}{n},\frac{1}{n},\ldots,\frac{1}{n})^T$
x_5	$(\frac{1}{10},\frac{1}{10},\frac{1}{10},\ldots,\frac{1}{10})^T$

These algorithms are implemented for dimensions n (1000, 2000, 5000, 7000, 12000). The stopping scale is $\|F(x_k)\| < 10^{-8}$. These algorithms are distinguished by their performance in (Iter): number of iterations, (Eval-F): number of function evaluations, (Time): CPU time in second and (Norm): the norm of approximation solution. The test problems are $F(x) = (f_1, f_2, f_3, \ldots, f_n)^T$ where $x = (x_1, x_2, x_3, \ldots, x_n)^T$, for $i = 1,2, \ldots, n$ *and* $\Omega = \mathbb{R}_+^n$ Information of test functions as shown in Table 2.

Winner w.r.t. number of iterations, FVAL, TIME and NORM as shown in Table 3, according to all the initial points that we chose, shows the number of times the new algorithm has succeeded (NDDL-CG) versus each other against (PDL-CG) by relying on the number of iterations and on the number of times the goal function is calculated and on the time taken for each implementation In addition to the base value. As for Table 4, it represents the total implementation results of the (new (NDDL-CG) and old (PDL-CG)) algorithms) for each starting point of the initial five points.

Table 2. Information of test functions [31-36]

Name of Functions	Details	Reference
F_1	$F_i(x) = 2\,x_i - \sin\lvert x_i\rvert$	[31]
F_2	$F_i(x) = x_i - \sin(x_i)$	[31]
F_3	$F_i(x) = e^{x_i} - 1$	[32]
F_4	$F_i(x) = \ln(\lvert x_i\rvert + 1) - \frac{x_i}{n}$	[33]
F_5	$F_i(x) = \min(\min(\lvert x_i\rvert, x_i^2), \max(\lvert x_i\rvert, x_i^3))$	[34]
F_6	$F_1(x) = x_1 - e^{\frac{\cos(x_1+x_2)}{n+1}}$ $F_i(x) = x_i - e^{\frac{\cos(x_{i+1}+x_i+x_{i-1})}{n+1}}, for\ i = 2,3,..,n-1$ $F_n(x) = x_n - e^{\frac{\cos(x_{n-1}+x_n)}{n+1}}$	[34]
F_7	$F_i(x) = \frac{i}{n} e^{x_i} - 1$	[31]

Name of Functions	Details	Reference
F_8	$F_1(x) = e^{x_1} - 1, F_i(x) = e^{x_i} - x_{i-1} - 1$	[33]
F_9	$F_i(x) = \sum_{i=1}^{n} \lvert x_i \rvert^i$	[35]
F_{10}	$F_i(x) = \sum_{i=1}^{n} \lvert x_i \rvert \ e^{-\sum_{i=}^{n} \sin(x_i^2)}$	[35]
F_{11}	$F_i(x) = \sum_{i=1}^{n} \lvert x_i \sin(x_i) + 0.1\,(x_i) \rvert$	[36]
F_{12}	$F_i(x) = \sum_{i=1}^{n} \lvert x_i \rvert^{i+1}$	[36]

Table3. Winner w.r.t. number of iterations, FVAL, TIME and NORM

Name of Variable	PDL-CG	NDDL-CG
	ITER / FVAL / TIME / NORM	ITER / FVAL / TIME / NORM
x_1	35 / 35 / 24 / 26	25 / 25 / 36 / 34
x_2	21 / 21 / 19 / 25	39 / 39 / 41 / 35
x_3	15 / 15 / 11 / 18	25 / 25 / 37 / 22
x_4	15 / 15 / 11 / 16	10 / 10 / 14 / 9
x_5	15 / 11 / 3 / 20	25 / 29 / 37 / 20
Total	101 / 97 / 68 / 105	124 / 128 / 165 /120

Table4. Total of functions for each initial points

Name of Variable	PDL-CG	NDDL-CG
	ITER / FVAL / TIME / NORM	ITER / FVAL / TIME / NORM
x_1	1658 / 3612 / 1431.865 / 2.96E-07	431 / 1136 / 931.698 / 1.26E-07
x_2	1758 / 3611 / 2441.866 / 2.96E-07	382 / 990 / 835.6486 / 1.11E-07
x_3	1201 / 2607 / 1583.853 / 1.52E-07	307 / 945 / 681.346 / 9.7E-08
x_4	135 / 680 / 1196.513 / 2.63E-08	74 / 173 / 347.9768 / 1.65E-08
x_5	1678 / 4987 / 2678.288 / 1.89E-07	226 / 652 / 458.5485 / 7.13E-08
Total	6430 / 15497 / 9332.385 / 9.59E-07	1420 / 3896 / 3255.2179 / 4.22E-07

Using Dolan and Mor´e [37] style, the following three figures are also for comparison between the two algorithms concerning the number of iterations, the number of times the function is calculated and the time is taken, which we calculated for the point x_5, Figure 1 shows the effect of the number of iterations on the two algorithms when switching and increasing dimensions. As for Figure 2, it is clear that the new algorithm is based on calculating the number of times the target function is better. Figure 3 shows the amount of time spent on the algorithms used in this work. As a conclusion, the figure shows us that the new algorithm is more efficient when compared to the old algorithm.

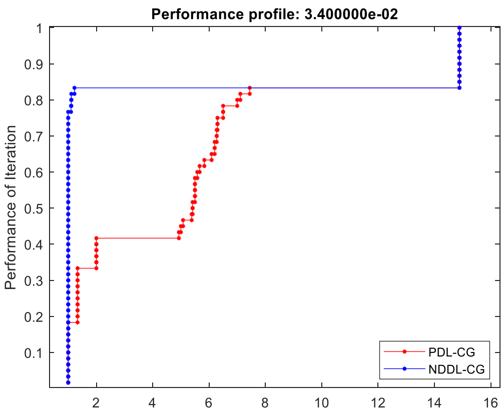

Figure 1. Performance of the two algorithms w.r.t. (Iter)

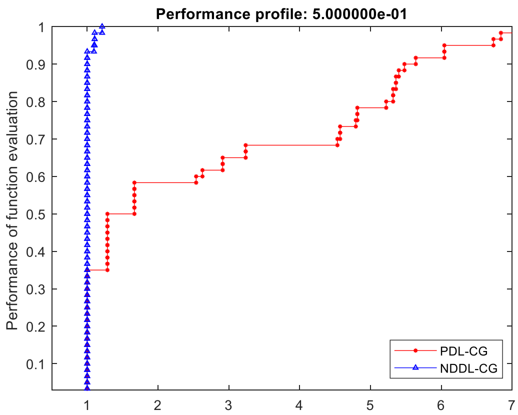

Figure 2. Performance of the two algorithms w.r.t. (Eval-F)

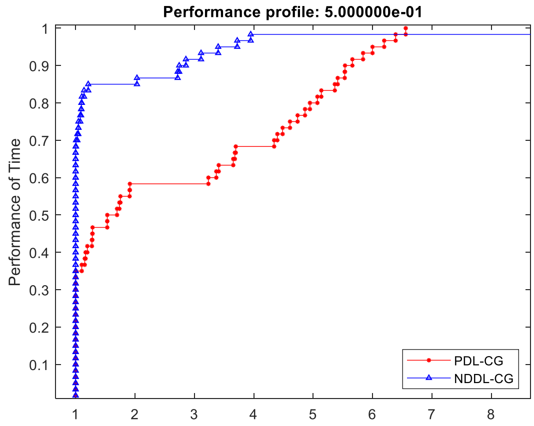

Figure 3. Performing the two algorithms w.r.t. (TIME)

5. CONCLUSION

From the results we conclude that the new algorithm (NDDL-CG) is more efficient than the old algorithm (PDL-CG) using most of the initial values when comparing its performance in changing dimensions. We also notice through the three drawings presented in the evaluation of previous results that the efficiency of the new algorithm increases with the increase in the number of dimensions and stability appears in some of the relevant results by calculating the goal function, and therefore the addition to the new algorithm (which contains the parameter of the penalty function) makes the new algorithm more appropriate than the algorithms Others are in the same field of work.

Acknowledgments

The research is supported by the College of Computer Sciences and Mathematics, University of Mosul, and College of Basic Education, Telafer University, Republic of Iraq. The authors declare that there are no conflicts of interest regarding this work.

REFERENCES

[1] J. Barzilai and J. M. Borwein, "Two-point step size gradient methods," *IMA Journal of numerical analysis*, vol. 8, no. 1, pp. 141-148, 1988.
[2] J. E. Dennis Jr and R. B. Schnabel, "Numerical methods for unconstrained optimization and nonlinear equations," *SIAM*, vol. 16, 1996.
[3] R. Z. Al-Kawaz, "Average Conjugate Gradient Method With Optimum Restart Powell For Nonlinear Function," *International Journal of Computer Science and Information Security (IJCSIS)***,** vol. 16, no. 8, 2018.
[4] A. Y. AL-Bayati and R. Z. AL-Kawaz, "Two New Extended PR Conjugate Gradient Methods for Solving Nonlinear Minimization Problems," *Journal of Education and Science*, vol. 27, no. 3, pp. 71-96, 2018.
[5] E. T. Hamed, H. I. Ahmed, and A. Y. Al-Bayati, "A New Hybrid Algorithm for Convex Nonlinear Unconstrained Optimization," *Journal of Applied Mathematics*, 2019.
[6] H. I. Ahmed, R. Z. Al-Kawaz and A. Y. Al-Bayati, "Spectral Three-Term Constrained Conjugate Gradient Algorithm for Function Minimizations," *Journal of Applied Mathematics*, 2019.
[7] 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*, 2020.
[8] A. Y. Al-Bayati and M. M. M. Ali, "New multi-step three-term conjugate gradient algorithms with inexact line searches," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 19, no. 3, pp. 1564-1573, 2020.
[9] A. A. Al-Arbo1 and R. Z. Al-Kawaz, "Implementation of combined new optimal cuckoo algorithm with a gray wolf algorithm to solve unconstrained optimization nonlinear problems," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 19, no. 3, pp. 1582-1589, 2020.
[10] R. Z. Al-Kawaz, A. Y. Al-Bayati, M. S. Jameel, "Interaction between updated FR-CG algorithms with optimal Cuckoo algorithm," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 19, no. 3, 2020.
[11] R. Z. Al-Kawaz, H. I. Ahmed, and A. Y. Al-Bayati, "An efficient modified Cuckoo search optimization with the rational barrier function.", *International Journal of Mathematics and Computer Applications Research (IJMCAR)*, vol. 8, no. 5, pp. 13-24, 2018.
[12] J. M. Ortega and W. C. Rheinboldt, "Iterative solution of nonlinear equations in several variables," *SIAM*, vol. 30, 1970.

[13] 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, Springer*, pp. 355-369, 1998.
[14] Y. H. Dai, M. Al-Baali, and X. Yang, "A positive Barzilai–Borwein-like stepsize and an extension for symmetric linear systems," *Numerical Analysis, and Optimization, Springer, Cham*, pp. 59-75, 2015.
[15] C. Wang and Y. Wang, "A superlinearly convergent projection method for constrained systems of nonlinear equations," *Journal of Global Optimization*, vol. 44, no. 2, pp. 283-296, 2009.
[16] J. Liu and S. Li, "Spectral DY-type projection method for nonlinear monotone systems of equations," *Journal of Computational Mathematics*, vol. 33, no. 4, pp. 341–355, 2015.
[17] S. Y. Liu, Y. Y. Huang and H. W. Jiao, "Sufficient descent conjugate gradient methods for solving convex constrained nonlinear monotone equations," *In Abstract and Applied Analysis*, *Hindawi*, 2014.
[18] 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.
[19] L. Zhang and W. Zhou, "Spectral gradient projection method for solving nonlinear monotone equations," *Journal of Computational and Applied Mathematics*, vol. 196, no. 2, pp. 478–484, 2006.
[20] 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, no. 1, pp. 33–46, 2007.
[21] L. Zheng, "A new projection algorithm for solving a system of nonlinear equations with convex constraints," *Bulletin of the Korean Mathematical Society***,** vol. 50, no. 3, pp. 823–832, 2013.
[22] Q. Li and D. H. Li, "A class of derivative-free methods for large-scale nonlinear monotone equations," *IMA Journal of Numerical Analysis*, vol. 31, no. 4, pp. 1625-1635, 2011.
[23] A. B. Abubakar, P. Kumam, and H. Mohammad, "A note on the spectral gradient projection method for nonlinear monotone equations with applications," *Computational and Applied Mathematics*, vol. 39, no. 129, pp. 1-35, 2020.
[24] M. Koorapetse and P. Kaelo, "Self-adaptive spectral conjugate gradient method for solving nonlinear monotone equations," *Journal of the Egyptian Mathematical Society*, vol. 28, no. 4, pp. 4, 2020.
[25] M. Ahookhosh, K. Amini and S. Bahrami, "Two derivative-free projection approaches for systems of large-scale nonlinear monotone equations," *Numerical Algorithms*, vol. 64, no. 1, pp. 21-42, 2013.
[26] Y. Ou and J. Li, "A new derivative-free SCG-type projection method for nonlinear monotone equations with convex constraints," *J. Appl. Math. Comput*, vol. 56, pp. 195–216, 2018.
[27] 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, pp. 745, 2019.
[28] 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.
[29] J. K. Liu and S. J. Li, "A projection method for convex constrained monotone non-linear equations with applications," *Computers, and Mathematics with Applications*, vol. 70, no. 10, pp. 2442-2453, 2015.
[30] M. Fatemi, "A new efficient conjugate gradient method for unconstrained optimization," *Journal of Computational and Applied Mathematics*, vol. 300, pp. 207-216, 2016.
[31] W. Zhou and D. Li, "Limited memory BFGS method for nonlinear monotone equations," *Journal of Computational Mathematics*, pp. 89-96, 2007.
[32] 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, no. 1, pp. 33-46, 2007.
[33] W. La Cruz, J. 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.
[34] W. La Cruz, "A spectral algorithm for large-scale systems of nonlinear monotone equations," *Numerical Algorithms*, vol. 76, no. 4, pp. 1109-1130, 2017.
[35] X. S. Yang, "Test Problems in Optimization, Engineering Optimization: An Introduction with Metaheuristic Applications," *John Wiley & Sons*, 2010.
[36] M. Jamil and X. S. Yang, "A literature survey of benchmark functions for global optimization problems," *Int. Journal of Mathematical Modelling and Numerical Optimisation*, vol. 4, no. 2, pp. 150–194, 2013.
[37] E. D. Dolan and J. J. Mor´e, "Benchmarking optimization software with performance profiles," *Mathematical programming*, vol. 91, no. 2, pp. 201-213, 2002.