AIMS Mathematics, 8(12): 30940–30962.
DOI: 10.3934/math.20231583
Received: 16 August 2023
Revised: 24 September 2023
Accepted: 13 October 2023
Published: 17 November 2023

http://www.aimspress.com/journal/Math

Research article

An efficient spectral minimization of the Dai-Yuan method with application to image reconstruction

Nasiru Salihu[1,2], Poom Kumam[1,3,*], Ibrahim Mohammed Sulaiman[4,5] and Thidaporn Seangwattana[6]

[1] Center of Excellence in Theoretical and Computational Science (TaCS-CoE), Fixed Point Research Laboratory, Fixed Point Theory and Applications Research Group, Faculty of Science, King Mongkut's University of Technology Thonburi (KMUTT), Bangkok 10140, Thailand

[2] Department of Mathematics, Faculty of Sciences, Modibbo Adama University, Yola 652105, Nigeria

[3] KMUTT-Fixed Point Research Laboratory, Room SCL 802, Science Laboratory Building, Department of Mathematics,Faculty of Science, King Mongkut's University of Technology Thonburi (KMUTT), Bangkok 10140, Thailand

[4] Institute of Strategic Industrial Decision Modelling, School of Quantitative Sciences, Universiti Utara Malaysia, Sintok, Kedah 06010, Malaysia

[5] Faculty of Education and Arts, Sohar University, Sohar 311, Oman

[6] Faculty of Science Energy and Environment, King Mongkut's University of Technology North Bangkok, Rayong Campus (KMUTNB), Rayong 21120, Thailand

* **Correspondence:** Email: poom.kum@kmutt.ac.th.

Abstract: In this paper, a spectral Dai and Yuan conjugate gradient (CG) method is proposed based on the generalized conjugacy condition for large-scale unconstrained optimization, in which the spectral parameter is motivated by some interesting theoretical features of quadratic convergence associated with the Newton method. Accordingly, utilizing the strong Wolfe line search to yield the step-length, the search direction of the proposed spectral method is sufficiently descending and converges globally. By applying some standard Euclidean optimization test functions, numerical results reports show the advantage of the method over some modified Dai and Yuan CG schemes in literature. In addition, the method also shows some reliable results, when applied to solve an image reconstruction model.

Keywords: unconstrained optimization; spectral conjugate gradient method; generalized conjugacy condition; Newton direction; global convergence; image reconstruction
Mathematics Subject Classification: 65K05, 90C06, 90C30, 90C47, 90C90

1. Introduction

The unconstrained optimization basically deals with finding the global maximum or minimum of a continuously differentiable function $f : \mathbb{R}^n \longrightarrow \mathbb{R}$. The general form of an unconstrained optimization problem is as follows:

$$\min f(x),\ x \in \mathbb{R}^n, \tag{1.1}$$

where the gradient $g(x) = \nabla f(x)$. Solving the complexity of this problem (1.1) would require the use of some symbolic aided algebra system and an effective numerical method that would be able to perform the necessary computations, plot the numerical results and manipulate the mathematical expression in an analytical form. One of the widely and most preferred methods considered for solving (1.1) is the nonlinear conjugate gradient (CG) method because of its robustness and ability to deal with large-scale problems [1,2]. The simplicity and efficiency of the CG algorithm has further motivated its application to numerous real-life problems including the feed forward training of neural networks [3], robotic motion control [4], signal recovery [5], regression analysis [1, 6], image deblurring [7, 8] and portfolio selection [9] among others.

Starting with an initial guess of $x_0 \in \mathbb{R}^n$, the CG method generates a sequence of iterates $\{x_k\}$ via the following formula:

$$x_{k+1} = x_k + \alpha_k d_k,\ \ k = 0, 1, 2, \ldots, \tag{1.2}$$

where α_k represents the step size that is computed along the search direction d_k [10]. Usually, a line search procedure is required to obtain the step size (α_k) which could either be an exact or inexact procedure. While the exact line search yields the exact minimizer for a given problem, the inexact scheme computes α_k such that it satisfies the Wolfe conditions at each iteration. The most commonly used Wolfe conditions includes the standard Wolfe condition that requires α_k to satisfy

$$f(x_k + \alpha_k d_k) \leq f(x_k) + \delta\alpha_k g_k^T d_k, \tag{1.3}$$

$$g(x_k + \alpha_k d_k)^T d_k \geq \sigma g_k^T d_k, \tag{1.4}$$

and the strong Wolfe condition is computed such that α_k satisfies (1.3) and

$$|g(x_k + \alpha_k d_k)^T d_k| \leq -\sigma g_k^T d_k, \tag{1.5}$$

where $0 < \delta < \frac{1}{2}$, $\delta < \sigma < 1$.

Another important component of the CG method is the search direction d_k. This component plays an important role in the performance and convergence analysis of any CG method. Also, the search direction distinguishes the classes of different CG algorithms including the spectral and three-term CG methods. The search direction d_k for the classical CG method is computed as follows:

$$d_0 = -g_0,\ d_{k+1} = -g_{k+1} + \beta_k d_k,\ \ \forall k \geq 1, \tag{1.6}$$

where the coefficient β_k characterizes the different CG formulas. Selection of the CG choice parameter and search direction is always crucial in the study of the unconstrained optimization because these

two components are responsible for the numerical performance and theoretical analysis of any CG method [11].

For any optimization method to fulfill the general optimization criteria and be able to use the line search procedure, it is required to possess the following descent property:

$$d_k^T g_k < 0 \quad \forall k. \tag{1.7}$$

Furthermore, a CG method is said to be sufficiently descending if there exists some constant $c > 0$ such that the following condition holds:

$$d_k^T g_k \leq -c\|g_k\|^2 \quad \forall k. \tag{1.8}$$

This condition (1.8) plays an important role in the convergence analysis of any nonlinear CG algorithm, which is always preferred over (1.7). Some of the earliest and known CG coefficients that possess this condition (1.8) include those developed by Fletcher and Revees (FR) [12], Dai and Yuan (DY) [13] and Fletcher (conjugate descent (CD)) [14], with formulas as follows:

$$\beta_k^{FR} = \frac{\|g_{k+1}\|^2}{\|g_k\|^2}, \quad \beta_k^{DY} = \frac{\|g_{k+1}\|^2}{d_k^T y_k}, \quad \beta_k^{CD} = -\frac{\|g_{k+1}\|^2}{d_k^T g_k}, \tag{1.9}$$

where $y_k = g_{k+1} - g_k$ and $\|\cdot\|$ denotes the ℓ_2 norm. This class of CG methods is characterized by simplicity and less memory requirements under different line search procedures. However, their numerical performance in practical computations is often affected by jamming phenomena. For instance, if any of the above CG algorithm generates a tiny step size from x_k to x_{k+1} and poor search direction and as a result if restart is not performed along the negative direction, then, it is likely that the subsequent step size and direction will also have poor performance [15].

Due to the challenges reported by the above category of CG algorithms, several studies have shown that the methods possess nice convergence properties (see [11, 16, 17]). To address the issue related to the computational performance, several studies involved constructing new CG formulas by either combining the methods in (1.9) with other efficient formulas or introducing new terms to the set of methods in (1.9) to improve the computational efficiency and their general structure [18].

In this study, we are interested in modifying the classical DY CG formula for solving optimization and image restoration problems. The proposed method introduces a new spectral parameter θ_k to scale the search direction d_k in (1.6) such that it satisfies the Newton search direction and the well-known D–L [19] conjugacy condition in Section 2. Section 3 demonstrates how the proposed method satisfies the descent condition and further proves the global convergence under suitable conditions. The preliminary computational results on a set of optimization functions are analyzed in Section 4 while Section 5 reports results related to real-world application problem. The last section summarizes the whole idea related to the study and presents a general conclusion.

2. Formulation of the spectral algorithm and motivation

The spectral CG methods have been proposed to improve the general structure of CG methods. The methods are based on the works of Raydan [20], Barzilai and Borwein [21] and Birgin and Martínez [22]. For some recent results on spectral CG methods, see [8,23–31]. Similarly, the standard conjugacy condition given by

$$y_k^T d_{k+1} = 0 \tag{2.1}$$

is crucial in the convergence analysis of these CG methods. The CG methods proposed with this structure (2.1) largely depend on exact line criteria for the choice of step size, α_k. However, this requirement is computationally expensive for large-scale problems. Therefore, denoting $s_k = x_{k+1} - x_k$, the selection of α_k uses its generalized form called the D–L conjugacy condition introduced in [19],

$$y_k^T d_{k+1} = -t g_{k+1}^T s_k, \;\; t > 0, \tag{2.2}$$

is usually preferable in the design of some nice CG algorithms. The parameter t, called the D-L parameter is capable of ensuring better convergence. Therefore, it must be carefully chosen.
To address some weaknesses associated with the DY method, its modification has been investigated recently. For instance, utilizing the strong Wolfe line search, Jiang and Jian [32] proposed the improved DY [13] CG parameter with the following structure:

$$\beta_k^{IDY} = r_k \frac{\|g_{k+1}\|^2}{d_k^T y_k}, \tag{2.3}$$

where $r_k = \frac{|g_{k+1}^T d_k|}{-g_k^T d_k}$. Motivated by this idea, Jian et al. [33] introduced a spectral parameter that yields the descent direction with

$$\theta_k^{JYJLL} = 1 + \frac{|g_{k+1}^T d_k|}{-g_k^T d_k}. \tag{2.4}$$

Since $\theta_k^{JYJLL} \geq 1$ it is obvious that d_k is a descent direction. Following the idea in [34], they proposed its conjugate parameter as follows:

$$\beta_k^{JYJLL} = \frac{\|g_{k+1}\|^2 - \frac{(g_{k+1}^T d_k)^2}{\|d_k\|^2}}{\max\{\|g_k\|^2, \; d_k^T y_k\}}. \tag{2.5}$$

However, the search direction including the method in (2.5) only satisfies the descent condition, (1.7) and it converges globally based on two cases of $\max\{\|g_k\|^2, \; d_k^T y_k\}$ with the standard Wolfe criteria respectively. According to this formulation, two other modified DY CG methods were proposed in [35] with the following forms:

$$\beta_k^{(1)} = \frac{\|g_{k+1}\|^2 - \frac{(g_{k+1}^T g_k)^2}{\|g_k\|^2}}{d_k^T y_k}, \tag{2.6}$$

$$\beta_k^{(2)} = \frac{\|g_{k+1}\|^2 - \frac{\|g_{k+1}\|}{\|g_k\|} g_{k+1}^T g_k}{\mu d_k^T y_k}. \tag{2.7}$$

Inspired by these modifications, Zhu et al. [36] suggested another DY modification namely, the DDY1 scheme which has the form of

$$\beta_k^{DDY1} = \begin{cases} \frac{\|g_{k+1}\|^2 - \frac{\mu_1 (g_{k+1}^T d_k)^2}{\|g_{k+1}\| \|g_k\| \|d_k\|^2} g_{k+1}^T g_k}{d_k^T y_k}, & g_{k+1}^T g_k \geq 0, \\ \frac{\|g_{k+1}\|^2 + \frac{\mu_1 (g_{k+1}^T d_k)^2}{\|g_{k+1}\| \|g_k\| \|d_k\|^2} g_{k+1}^T g_k}{d_k^T y_k}, & g_{k+1}^T g_k < 0. \end{cases} \tag{2.8}$$

Nonetheless, the method has similar theoretical characteristics with the scheme described by (2.5). Thus, in order to improve the general structure and establish sufficient descent, as given by (1.8) for the DY method, we scale the search direction d_k in (1.6) with a spectral parameter θ_k such that it satisfies the well-known D–L conjugacy condition (2.2), as well as the Newton search direction in the following form

$$d_{k+1} = -\theta_k g_{k+1} + \beta_k d_k, \quad \forall k \geq 1. \tag{2.9}$$

Pre-multiplying by y_k^T, we obtain

$$y_k^T d_{k+1} = -\theta_k y_k^T g_{k+1} + \beta_k^{DY} y_k^T d_k.$$

Using the above relation with (2.2) gives

$$-t s_k^T g_{k+1} = -\theta_k y_k^T g_{k+1} + \beta_k^{DY} y_k^T d_k.$$

Re- arranging implies that

$$\theta_k^{SDL} = \frac{t s_k^T g_{k+1}}{y_k^T g_{k+1}} + \beta_k^{DY} \frac{y_k^T d_k}{y_k^T g_{k+1}}. \tag{2.10}$$

Similarly, in some neighborhood of the minimizer, if the current point x_{k+1} is close enough to a local minimizer and the objective function behaves like a quadratic function, then the optimal search direction to follow is the Newton direction:

$$d_{k+1} = -\nabla^2 f(x_{k+1})^{-1} g_{k+1}, \tag{2.11}$$

where $\nabla^2 f(x_{k+1})$ is the Hessian matrix of f. Thus, the Newton method requires the Hessian matrix, i.e, the second derivative information to update (2.11), which provides a nice convergence rate. Motivated by its quadratic convergence property, we assume that $\nabla^2 f(x_{k+1})$ exists at every iteration and satisfies the conditions of a suitable secant equation; for instance,

$$\nabla^2 f(x_{k+1}) s_k = y_k. \tag{2.12}$$

Now, equating (2.9) with (2.11) gives

$$-\nabla^2 f(x_{k+1})^{-1} g_{k+1} = -\theta_k g_{k+1} + \beta_k^{DY} d_k.$$

Pre-multiplying by y_k^T and using the secant equation (2.12), we get

$$\begin{aligned} -\nabla^2 f(x_{k+1})^{-1} g_{k+1} &= -\theta_k g_{k+1} + \beta_k^{DY} d_k \\ -s_k^T g_{k+1} &= -\theta_k s_k^T \nabla^2 f(x_{k+1}) g_{k+1} + \beta_k^{DY} s_k^T \nabla^2 f(x_{k+1}) d_k \\ -s_k^T g_{k+1} &= -\theta_k y_k^T g_{k+1} + \beta_k^{DY} y_k^T d_k. \end{aligned}$$

After some simple simplification, we obtain

$$\theta_k^{SNM} = \frac{s_k^T g_{k+1}}{y_k^T g_{k+1}} + \beta_k^{DY} \frac{y_k^T d_k}{y_k^T g_{k+1}}. \tag{2.13}$$

Remark 2.1. *Observe that, if* $t = 1$*, then the parameter* $\theta_k^{SDL} = \theta_k^{SNM}$*, which implies that, the spectral search direction* d_{k+1} *does not only satisfy the generalized D-L conjugacy condition it also takes advantage of the nice convergence property of the Newton direction. However, for a sufficient descent condition to hold, we always select* $t > 1$ *in this paper.*

To achieve the sufficient descent of the search direction described by (2.9), we suggest the following modification of (2.10) as follows:

$$\theta_k^{SNM1} = \frac{ts_k^T g_{k+1}}{\max\{|y_k^T g_{k+1}|,\ d_k^T y_k\}} + \beta_k^{DY} \frac{y_k^T d_k}{\max\{|y_k^T g_{k+1}|, d_k^T y_k\}}. \tag{2.14}$$

Thus, if $|y_k^T g_{k+1}| > d_k^T y_k$, then (2.14) reduces (2.10). Hence, for $d_k^T y_k > |y_k^T g_{k+1}|$, we obtain another spectral parameter:

$$\theta_k^{SNM2} = \frac{ts_k^T g_{k+1}}{d_k^T y_k} + \beta_k^{DY} \frac{y_k^T d_k}{d_k^T y_k}. \tag{2.15}$$

Combining (2.10) and (2.15) implies that the optimal θ_k can be computed as follows:

$$\theta_k^{NSDY} = \begin{cases} \theta_k^{SDL}, & if\ \ |y_k^T g_{k+1}| > d_k^T y_k, \\ \theta_k^{SNM2}, & \text{otherwise.} \end{cases} \tag{2.16}$$

Now, we describe the new spectral DY (NSDY) algorithm as follows:

Algorithm 1: NSDY

Step 1: Apply $x_0 \in \mathbb{R}^n$ and the parameters $0 < \delta < \sigma \leq 1$. Set $d_0 = -g_0$, $\alpha_0 = 1$.

Step 2: Verify the convergence: If $\|g_k\| \leq \epsilon$, then stop. Otherwise , proceed to the next step.

Step 3: Select $\alpha_k > 0$ such that (1.3) and (1.5) are satisfied.

Step 4: Compute θ_k^{NSDY} using (2.16).

Step 5: Compute β_k^{DY} from (1.9) and d_k using (2.9).

Step 6: Update the next iteration from Step 2.

Theorem 2.2. *Let* θ_k *be given by* (2.16) *with* $0 < \alpha_k \leq \sigma < 1$*. Suppose that the NSDY algorithm generates sequences* $\{g_k\}$ *and* $\{d_k\}$*, where* α_k *is determined by strong Wolfe rules* (1.3)–(1.5)*; then, there exists a constant* $\rho > 0$ *for which the condition*

$$d_{k+1}^T g_{k+1} \leq -\rho \|g_{k+1}\|^2, \ \ \forall k \geq 0. \tag{2.17}$$

holds.

The criterion (2.17) is called the sufficient descent condition, and it ensure that the direction described by (2.9) is indeed sufficient for the minimizer.

Proof. The proof is achieved by mathematical induction. Initially, for $k = 0$ it follows easily that $g_0^T d_0 = -\|g_0\|^2 \leq -\rho \|g_0\|^2$. Suppose that (2.17) holds for index k, that is, $d_k^T g_k \leq -\rho \|g_k\|^2$. We now show for $k + 1$. Now, according to the strong Wolfe criterion (1.3), it holds that

$$|g_{k+1}^T d_k| \leq -\sigma g_k^T d_k,$$

i.e.,

$$d_k^T y_k = d_k^T g_{k+1} - d_k^T g_k \geq -(1-\sigma) d_k^T g_k. \tag{2.18}$$

Then

$$\frac{|g_{k+1}^T d_k|}{|d_k^T y_k|} \leq \frac{\sigma}{1-\sigma}. \tag{2.19}$$

Pre-multiplying (2.9) by g_{k+1}^T, we have

$$g_{k+1}^T d_{k+1} = -\theta_k \|g_{k+1}\| + \beta_k^{DY} g_{k+1}^T d_k. \tag{2.20}$$

Using (2.14) with $s_k = \alpha_k d_k$ gives

$$\begin{aligned}
\theta_k^{SNM1} &= \frac{t s_k^T g_{k+1}}{\max\{|y_k^T g_{k+1}|,\ d_k^T y_k\}} + \beta_k^{DY} \frac{y_k^T d_k}{\max\{|y_k^T g_{k+1}|, d_k^T y_k\}} \\
&\leq \frac{t s_k^T g_{k+1}}{|d_k^T y_k|} + \beta_k^{DY} \frac{y_k^T d_k}{|d_k^T y_k|} \\
&\leq \frac{t\alpha_k |d_k^T g_{k+1}|}{|d_k^T y_k|} + \beta_k^{DY} \frac{y_k^T d_k}{|d_k^T y_k|}.
\end{aligned} \tag{2.21}$$

Obtaining β_k^{DY} from (1.9) and applying it with (2.21) and (2.20), we get

$$\begin{aligned}
g_{k+1}^T d_{k+1} &\leq -\Big(\frac{t\alpha_k |d_k^T g_{k+1}|}{|d_k^T y_k|} + \beta_k^{DY} \frac{y_k^T d_k}{|d_k^T y_k|}\Big)\|g_{k+1}\|^2 + \beta_k^{DY} g_{k+1}^T d_k. \\
&= -\Big(\frac{t\alpha_k |g_{k+1}^T d_k|}{|d_k^T y_k|} + \frac{\|g_{k+1}\|^2}{|d_k^T y_k|}\Big)\|g_{k+1}\|^2 + \frac{\|g_{k+1}\|^2}{d_k^T y_k} g_{k+1}^T d_k. \\
&\leq -\Big(\frac{t\alpha_k |g_{k+1}^T d_k|}{|d_k^T y_k|} + \frac{\|g_{k+1}\|^2}{|d_k^T y_k|}\Big)\|g_{k+1}\|^2 + \frac{|g_{k+1}^T d_k|}{|d_k^T y_k|}\|g_{k+1}\|^2 \\
&\leq -\Big(\frac{\sigma t\alpha_k}{1-\sigma} + \frac{\|g_{k+1}\|^2}{|d_k^T y_k|}\Big)\|g_{k+1}\|^2 + \frac{\sigma}{1-\sigma}\|g_{k+1}\|^2
\end{aligned} \tag{2.22}$$

where the second to the last and last inequalities follow (2.19) respectively. Now, from (2.18), we get

$$\frac{1}{d_k^T y_k} \leq \frac{1}{-(1-\sigma) d_k^T g_k}.$$

Combining this with (2.17), we get

$$\frac{1}{d_k^T y_k} \leq \frac{1}{(1-\sigma)\rho\|g_k\|^2}. \tag{2.23}$$

The result by the last inequality follows from the induction hypothesis. Finally, using (2.22) and (2.23), we conclude that

$$g_{k+1}^T d_{k+1} \leq -\Big(\frac{\sigma t\alpha_k}{(1-\sigma)} + \frac{\|g_{k+1}\|^2}{(1-\sigma)\rho\|g_k\|^2} - \frac{\sigma}{(1-\sigma)}\Big)\|g_{k+1}\|^2.$$

$$\leq -\left(\frac{\sigma t\alpha_k}{(1-\sigma)} - \frac{\sigma}{(1-\sigma)}\right)\|g_{k+1}\|^2.$$

Since $\frac{\|g_{k+1}\|^2}{(1-\sigma)\rho\|g_k\|^2} > 0$, we have

$$g_{k+1}^T d_{k+1} \leq -\left(\frac{\sigma t\alpha_k}{(1-\sigma)} - \frac{\sigma}{(1-\sigma)}\right)\|g_{k+1}\|^2.$$

Denoting $\rho = \left(\frac{\sigma t\alpha_k}{(1-\sigma)} - \frac{\sigma}{(1-\sigma)}\right)$, with $t \geq 1.2$, $\alpha_k \in (0, 0.9]$ and $\sigma \in (0, 0.9]$ the required result is achieved. □

3. Convergence analysis

The convergence analysis requires the following presumptions.

Assumption 3.1.

1. *Given an initial point x_0, the function $f(x)$ is bounded below on the level set $\eta = \{x \in \mathbb{R}^n : f(x_0) \geq f(x)\}$.*
2. *Denote Γ as some neighborhood of η, and the function f is smooth with its gradient that is Lipschitz-continuous and satisfying*

$$\|g(x) - g(y)\| \leq L\|x - y\|, \quad \forall x, y \in \Gamma, \quad L > 0. \tag{3.1}$$

Note that these assumptions, imply that

$$\|g(x)\| \leq \gamma, \quad \forall x \in \eta, \quad \gamma > 0, \tag{3.2}$$

$$\|x - y\| \leq b, \quad \forall x, y \in \eta, \quad b > 0. \tag{3.3}$$

The following lemma was taken from [17] and is very crucial in our analysis.

Lemma 3.2. *Suppose that $\{x_k\}$ and $\{d_k\}$ are sequences generated by the NSDY algorithm, where the spectral search direction d_k is descending and α_k satisfies the strong Wolfe conditions; then,*

$$\sum_{k\geq 0}^{\infty} \frac{(g_k^T d_k)^2}{\|d_k\|^2} < +\infty. \tag{3.4}$$

Theorem 3.3. *If Assumption* (3.1) *holds and the sequence of iterations $\{x_k\}$ is produced by the NSDY algorithm, then*

$$\liminf_{k\to\infty} \|g_k\| = 0. \tag{3.5}$$

Proof. If (3.5) does not hold, then there exists some constant $r > 0$ so that

$$\|g_k\| > r. \tag{3.6}$$

Claim: The search direction defined by (2.9) is bounded, i.e., there exists a constant $P > 0$ such that

$$\|d_{k+1}\| \leq P, \quad \forall k \geq 0. \tag{3.7}$$

To proceed with the proof of this claim by induction, we need to show that $|\beta_k^{DY}|$ and $|\theta_k|$ are bounded below. According to (2.18), it yields

$$\frac{1}{d_k^T y_k} \leq \frac{1}{-(1-\sigma)d_k^T g_k}.$$

Combining this with (2.17), we get

$$\frac{1}{d_k^T y_k} \leq \frac{1}{(1-\sigma)\rho\|g_k\|^2}. \tag{3.8}$$

Therefore, obtaining β_k^{DY} from (1.9) and applying it with (3.2), (3.6) and (3.8), we have

$$|\beta_k^{DY}| \leq \frac{\|g_{k+1}\|^2}{|d_k^T y_k|} \leq \frac{\gamma^2}{\rho(1-\sigma)r^2} = E. \tag{3.9}$$

Next, from (2.14), (3.2), (3.8) and (3.9), we obtain

$$\begin{aligned}
|\theta_k| &\leq \left| \frac{t s_k^T g_{k+1}}{\max\{|y_k^T g_{k+1}|,\ d_k^T y_k\}} + \beta_k^{DY} \frac{y_k^T d_k}{\max\{|y_k^T g_{k+1}|,\ d_k^T y_k\}} \right| \\
&\leq \left| \frac{t s_k^T g_{k+1}}{d_k^T y_k} + \beta_k^{DY} \right| \\
&\leq \frac{t\|s_k\|\|g_{k+1}\|}{|d_k^T y_k|} + E \\
&\leq \frac{tb\gamma}{\rho(1-\sigma)r^2} + E = M.
\end{aligned} \tag{3.10}$$

Now, for $k = 0$, we get that $d_1 = -\theta_1 g_1 + \beta_1 d_0$ from (2.9), which implies that $d_1 = -\theta_1 g_1 - \beta_1 g_0$, since $d_0 = -g_0$; this yields

$$\begin{aligned}
\|d_1\| &\leq |\theta_1|\|g_1\| + |\beta_1|\|g_0\| \\
&\leq \gamma + E\gamma = \gamma^*,
\end{aligned}$$

that is, the claim (3.7) holds for $k = 0$. Next we assume that the claim (3.7) is true for k, that is, $\|d_k\| \leq P$. To show that it is true for $k+1$, consider the search direction described by (2.9).

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

Thus, using (3.2), (3.9) and (3.10), we obtain

$$\begin{aligned}
\|d_{k+1}\| &\leq |\theta_k|\|g_{k+1}\| + |\beta_k|\|d_k\| \\
&\leq M\gamma + EP;
\end{aligned}$$

therefore, the claim also holds for $k+1$. Now since (3.7) holds for all values of k, then we have

$$\frac{1}{\|d_k\|} \geq \frac{1}{P}, \quad P > 0. \tag{3.11}$$

From the above inequality, it follows that

$$\sum_{k=0}^{\infty} \frac{1}{\|d_k\|^2} = +\infty. \tag{3.12}$$

Accordingly, from (2.17), (3.4) and (3.6), it follows that

$$\rho^2 r^4 \sum_{k\geq 0}^{\infty} \frac{1}{\|d_k\|^2} \leq \sum_{k\geq 0}^{\infty} \frac{\rho^2 \|g_k\|^4}{\|d_k\|^2} \leq \sum_{k\geq 0}^{\infty} \frac{(g_k^T d_k)^2}{\|d_k\|^2} < +\infty \tag{3.13}$$

Therefore, this implies that

$$\sum_{k=0}^{\infty} \frac{1}{\|d_k\|^2} < +\infty. \tag{3.14}$$

It is obvious that, (3.12) and (3.14) cannot hold concurrently. Thus, (3.5) must hold. □

4. Numerical results

In this part, we present a numerical comparison of the NSDY method versus the classical DY method and three other modifications of the DY method. The evaluations are based on a number of test functions considered from [37, 38]. For the computation, we consider the dimension ranging from 2 to 100,000 as presented in Tables 1 and 2. To analyze the results further, the following specifications are considered for the algorithms:

- JYJLL algorithm of Jian et al. [33] that uses (2.4) and (2.5) for the search direction described by (2.9) and $\delta = 0.01$ and $\sigma = 0.1$ in (1.3) and (1.4) respectively.
- IDY algorithm of Jiang and Jian [32] that uses (2.3) for the search direction described by (1.6) and $\delta = 0.01$ and $\sigma = 0.1$ in (1.3) and (1.4) respectively.
- DDY1 algorithm of Zhu et al. [36] that uses (2.8) for the search direction described by (1.6) with $\mu_1 = 0.4$ and $\delta = 0.01$ and $\sigma = 0.1$ in (1.3) and (1.4) respectively.
- DY algorithm of Dai and Yuan [13] with $\beta_k^{DY} = \frac{\|g_{k+1}\|^2}{d_k^T y_k}$ for the search direction described by (1.6) and $\delta = 0.01$ and $\sigma = 0.1$ in (1.3) and (1.4) respectively.

The MATLAB R2022a codes for the numerical experiment were run on a Dell Core i7 laptop with 16 GB RAM and a 2.90 GHz CPU. We set $\delta = 10^{-3}$ and $\sigma = 0.9$ in (1.3) and (1.5) for the NSDY method and $\|g_k\| \leq 10^{-6}$ as the stopping condition for all schemes. We also use the symbol $***$ to indicate a situation in which the stopping condition is not met. The results from the computational experiments based on number of iterations (NI), number of function evaluations (FE), and CPU time (ET) is presented at the following link `https://acrobat.adobe.com/link/review?uri=urn:aaid:scds:US:ed485b92-05e2-40f3-a1f2-6e159858c515`.

We will employ the performance profiling technique to interpret and discuss the performance of the methods examined. Let P be the collection of n_p test problems and S be the set of n_s solvers used in the comparison. The performance profiles thus constitute the performance metric for a problem $p \in P$ and a solver $s \in S$, which denote either NI, FE or ET relative to the other solvers $s \in S$ on a set of

problems $p \in P$. Then, Dolan and Moré [39] described the measure of performance ratio to compare and evaluate the performance:

$$r_{p,s} = \frac{f_{p,s}}{\min\left\{f_{p,s} : s \in S \text{ and } p \in P\right\}}.$$

Thus, the best method is indicated by the performance profile plot at the top curve. The experiments can therefore be interpreted graphically by using Figures 1–3 based on numerical performance.

The interpretation of Figure 1 for the value τ chosen within the $0 < \tau < 0.5$ interval shows that the NSDY method solved the test problems and won in 98%/38% to be the best, followed by the JYJLL method with 95%/45%, whereas the DY, IDY and DDY1 methods solved and won in 96%/25%, 90%/37% and 55%/20% of the given problems respectively. Accordingly, if we increase the τ to an interval $\tau \geq 1$, the NSDY method is the best with 68% accuracy, whereas the JYJLL, IDY, DY and DDY1 methods have 66%, 60%, 55% and 40% accuracy, respectively. Similarly, the comparison among the five schemes based on interpretation of Figure 2 shows that the NSDY, DY, JYJLL, IDY and DDY1 methods solved the problems and won in 98%/12%, 96%/28%, 94%/54%, 90%/30% and 57%/14% respectively for the value τ chosen within $0 < \tau < 0.5$. However, increasing the τ value to the interval $\tau \geq 6$ reveals that the NSDY method wins when solving 97% of the test problems compared to JYJLL, IDY, DY and DDY1 methods with 90%, 88%, 96%, and 52% of problems solved to the best respectively. Finally, Figure 3 also shows that for the value τ chosen within $0 < \tau < 0.5$, NSDY, DY, JYJLL, IDY and DDY1 methods solved and won in 98%/33%, 96%/08%, 94%/38%, 90%/22% and 58%/05% problems, respectively. Alternatively, taking the value of τ in the interval $\tau \geq 6$ reveals that the NSDY method wins when solving 97% of the test problems compared to the JYJLL, IDY, DY and DDY1 methods with 93%, 88%, 92%, and 52% of problems solved to the best, respectively. Therefore, interpretations of Figures 1–3 indicate that the NSDY method is more preferable than other CG methods.

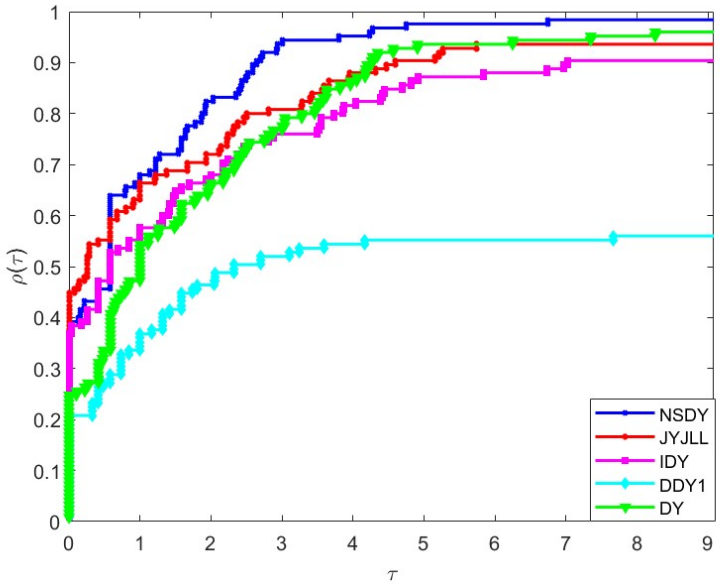

Figure 1. NI performance profile for the methods.

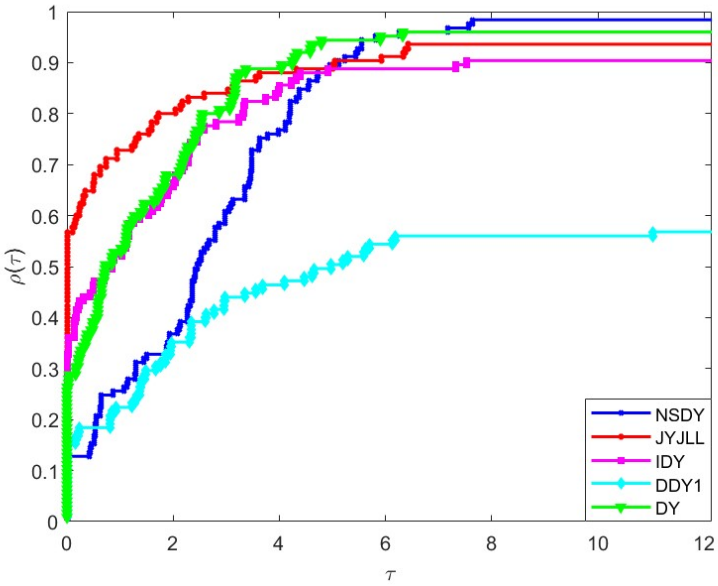

Figure 2. FE performance profile for the methods.

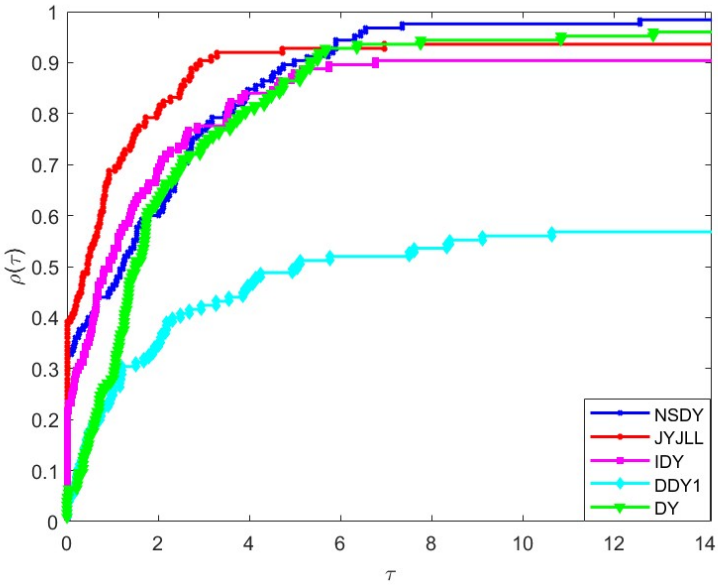

Figure 3. ET performance profile for the methods.

Table 1. List of test problems part 1/2.

NO	Problem	Dim	NO	Problem	Dim
1	EXTENDED PENALTY	100	43	DIAGONAL 1	100
2	EXTENDED PENALTY	10000	44	HAGER	500
3	EXTENDED PENALTY	20000	45	HAGER	300
4	EXTENDED MARATOS	2	46	HAGER	5000
5	DIAGONAL 5	10	47	HAGER	1000
6	DIAGONAL 5	50000	48	ZIRILLI	2
7	DIAGONAL 5	100000	49	RAYDAN 1	100
8	TRECANNI	2	50	RAYDAN 1	10
9	TRECANNI	2	51	RAYDAN 1	10
10	QUADRATIC PENALTY 1	4	52	RAYDAN 1	50
11	QUADRATIC PENALTY 1	100	53	RAYDAN 2	10000
12	QUADRATIC PENALTY 1	1000	54	RAYDAN 2	50000
13	QUADRATIC PENALTY 2	500	55	RAYDAN 2	100000
14	QUADRATIC PENALTY 2	100	56	FLETCHCER	10
15	QUADRATIC PENALTY 2	10	57	FLETCHCER	1000
16	QUADRATIC FUNCTION 1	100	58	FLETCHCER	50000
17	QUADRATIC FUNCTION 1	10	59	DIAGONAL 3	2
18	QUADRATIC FUNCTION 1	10	60	DIAGONAL 3	10
19	QUADRATIC FUNCTION 2	50	61	EXTENDED DENSCHN B	100
20	QUADRATIC FUNCTION 2	1000	62	EXTENDED DENSCHN B	5000
21	QUADRATIC FUNCTION 2	5000	63	EXTENDED DENSCHN B	10000
22	POWER	2	64	DIAGONAL 6	10000
23	POWER	2	65	DIAGONAL 6	5000
24	ZETTL	2	66	DIAGONAL 6	10000
25	DIAGONAL 2	1000	67	DIAGONAL 6	50000
26	DIAGONAL 2	10000	68	DIAGONAL 4	1000
27	DIAGONAL 2	50000	69	DIAGONAL 4	10000
28	TEST	3	70	DIAGONAL 4	100000
29	TEST	3	71	DIAGONAL 7	10
30	SUM OF SQUARES	100	72	DIAGONAL 7	100
31	SUM OF SQUARES	2000	73	DIAGONAL 7	100
32	SUM OF SQUARES	5000	74	DIAGONAL 8	100
33	SHALLOW	1000	75	DIAGONAL 8	500
34	QUARTIC	100	76	DIAGONAL 9	10
35	QUARTIC	1000	77	DIAGONAL 9	100
36	QUARTIC	5000	78	DENSCHN A	3000
37	QUARTIC	10000	79	DENSCHN A	15000
38	MATYAS	2	80	DENSCHN C	1000
39	MATYAS	2	81	DENSCHN C	10000
40	DIAGONAL 1	10	82	EXTENDED BLOCK DIAGONAL 1	10
41	DIAGONAL 1	100	83	EXTENDED BLOCK DIAGONAL 1	100
42	DIAGONAL 1	10	84	EXTENDED BLOCK DIAGONAL 1	1000

Table 2. List of test problems part 2/2.

NO	Problem	Dim
85	HIMMELBLAU	1000
86	HIMMELBLAU	10000
87	HIMMELBLAU	50000
88	HIMMELBLAU	10000
89	DQDRTIC	1000
90	DQDRTIC	10000
91	DQDRTIC	100
92	DQDRTIC	10
93	QUARTICM	1000
94	QUARTICM	10000
95	LINEAR PERTURBED	5000
96	LINEAR PERTURBED	10000
97	LINEAR PERTURBED	20000
98	TWH	2
99	ENGVAL1	2
100	ENGVAL1	2
101	ENGVAL8	4
102	ENGVAL8	2
103	DENSCHN F	1000
104	DENSCHN F	10000
105	DENSCHN F	50000
106	ARWHEAD	10
107	ARWHEAD	100
108	ARWHEAD	500
109	SIX HUMP	2
110	PRICE4	2
111	PRICE4	2
112	ZIRILLI	2
113	ZIRILLI	2
114	EXTENDED HIMMELBLAU	500
115	EXTENDED HIMMELBLAU	1000
116	EXTENDED HIMMELBLAU	2000
117	ROTATED ELLIPSE	2
118	ROTATED ELLIPSE	2
119	EL-ATTAR-VIDYASAGAR-DUTTA	2
120	EL-ATTAR-VIDYASAGAR-DUTTA	2
121	EXTENDED HIEBERT	2
122	EXTENDED TRIDIAGONAL 1	100
123	EXTENDED TRIDIAGONAL 1	500
124	THUMP	2
125	THUMP	2

5. Application of the NSDY algorithm in restoration of corrupted images

The CG method is among the most efficient minimization algorithms used for faster optimization of both linear and non-linear systems because of its rapid solution for large-scale problems. Recently, the

CG iterative scheme has been widely considered for solving real-world application problems because of its fewer iterations and less computational resources required to optimize a given problem. Some of the most relevant problems solved by the CG method includes the feed forward training of neural networks [3], the problem of motion control of robotic manipulators [8], regression analysis [1,6], and the restoration of corrupted images [8,9].

Image restoration is among the family of inverse problems used to obtain some highly-quality images from those images that have possibly been corrupted during the data acquisition process. Choosing an appropriate algorithm that is capable of restoring these corrupted images is very crucial because knowledge of the degradation system has to be considered in order to achieve better quality images. In this study, we consider restoring some images which that include a building (512×512) and camera (512×512). These images were possibly corrupted by salt-and-pepper impulse noise during the data acquisition or storage process. For the purpose of this study, the following metrics have been employed to measure the quality of the restored images and evaluate the performance of the algorithms. These metrics include the following

- peak signal-to-noise ratio (PSNR),
- relative error (RelErr),
- CPU time.

The PSNR has widely been used to measure the perceptual quality of restored images because, it calculates the ratio between the power of corrupting noises and the maximum possible power of a signal affecting the fidelity of the representation. This metric is very crucial when evaluating the quality of both corrupted and restored images. It is important to note that a method with higher PSNR will produce images with better quality [40]. The PSNR can be obtained as follows

$$\begin{aligned} PNSR &= 10 \cdot \log_{10}\left(\frac{MAX_I^2}{MSE}\right) \\ &= 20 \cdot \log_{10}\left(\frac{MAX_I}{\sqrt{MSE}}\right) \\ &= 20 \cdot \log_{10}(MAX_I) - 10 \cdot \log_{10}(MSE) \end{aligned} \tag{5.1}$$

where MSE is used to measure the average pixel differences of the complete images and MAX_I represents the maximum pixel value of the images. Also, an algorithm with higher MSE values will produce greater difference between the original and corrupted images. The MSE values can be obtained via:

$$MSE = \frac{1}{N}\sum\sum\left(E_{i,j} - o_{i,j}\right)^2, \tag{5.2}$$

where N denotes image size, E is the edge image and o represents the original image.

5.1. Image restoration problem

Consider the following image restoration problem:

$$\mathcal{H}(u) = \sum_{(i,j)\in G}\left\{\sum_{(m,n)\in T_{i,j}/G}\phi_\alpha(u_{i,j} - \xi_{m,n}) + \frac{1}{2}\sum_{(m,n)\in T_{i,j}\cap G}\phi_\alpha(u_{i,j} - u_{m,n})\right\},$$

where the edge-preserving potential function $\phi_\alpha(t) = \sqrt{t^2 + \alpha}$ for some constant α whose value is given as 1.

The above problem is transformed into the following optimization problem [41]

$$\min \mathcal{H}(u),$$

where x is the original $M \times N$ pixel image. Also, G denotes the following index set of noise candidates x:

$$G = \{(i, j) \in Q | \bar{\xi}_{ij} \neq \xi_{ij},\ \xi_{ij} = s_{\min} \text{ or } s_{\max}\}, \tag{5.3}$$

where $i, j \in Q = \{1, 2, \cdot, M\} \times \{1, 2, \cdot, N\}$ its neighborhood is defined as

$$T_{i\ j} = \{(i,\ j-1), (i, j+1), (i-1, j), (i+1, j)\}. \tag{5.4}$$

From (5.3), the maximum and minimum values of the noisy pixel are $s_{\max}$ and $s_{\min}$. Also, ξ is the observed noisy corrupted image whose adaptive median filter is denoted as $\bar{\xi}$.

Next, we present the computational performance of the proposed algorithm in Tables 3, 4, and 5. The performance was compared with that of other similar algorithms with the same characteristics in terms of the PNSR, RelErr, and CPU time to demonstrate the robustness of the algorithm. All computations were carried out under the conditions of the strong Wolfe line search with the restoration noise degrees set as 30, 50 and 80, respectively.

From the results presented in the tables above, it is obvious that all of the methods are very competitive. Starting with the CPU time of the camera image, it can be observed that at 30% noise degree, the CPU time for the proposed method is lower than that of DDY1 and IDY algorithms but higher than that of DY and JYJLL methods. However, at the 50% noise degree, the proposed algorithm outperformed all of the other methods, achieving the shortest CPU time. Also, at the 80% noise degree, our method was superior to all other algorithms except the IDY algorithm which produced the shortest CPU time. Similarly, by considering the Performance of the new method in terms of restoring the building corrupted image also shows that the method is very competitive. Furthermore, overall performance analysis of the methods (see Tables 4 and 5) in terms of the PSNR and RelErr shows that the proposed method performed better because its produced higher values for PSNR and RelErr in most of the noise degrees considered. Based on the previous discussion, the higher the PNSR and RelErr values, the better the quality of the output images. For the graphical representation of the results, we refer the reader to Figures 4 and 5. These results were obtained for the 30%, 50%, and 80% noise degrees, respectively.

Based on Tables 3, 4 and 5 and Figures 4 and 5, we can conclude that the proposed method is more efficient and robust because it restored most of the corrupted images with high accuracy than other existing methods.

Table 3. Image restoration outputs for NSDY, DY, DDY1, JYJLL, and IDY in terms of CPU time (CPUT).

METHOD		NSDY	DY	DDY1	JYJLL	IDY
IMAGE	NOISE	CPUT	CPUT	CPUT	CPUT	CPUT
CAMERA	30%	48.2371	**46.2840**	48.3230	46.6328	74.4926
	50%	**76.4981**	112.2052	78.4817	76.7701	109.4473
	80%	170.6165	261.6915	184.9280	174.2533	**141.6278**
BUILDING	30%	60.47630	61.8992	64.5338	**46.4718**	98.3261
	50%	143.6863	147.1558	104.1381	**82.7440**	144.3797
	80%	174.7825	187.3404	**159.8783**	191.3842	187.2095

Table 4. Image restoration outputs for NSDY, DY, DDY1, JYJLL, and IDY in terms of RelErr.

METHOD		NSDY	DY	DDY1	JYJLL	IDY
IMAGE	NOISE	RelErr	RelErr	RelErr	RelErr	RelErr
CAMERA	30%	1.2046	1.0732	**1.2665**	1.1387	1.1129
	50%	1.6940	**1.9179**	1.6558	1.5858	1.6827
	80%	**3.6860**	3.5995	3.2609	3.0341	3.0700
BUILDING	30%	1.4926	1.4439	1.4831	1.3775	**1.5324**
	50%	2.5823	2.6177	2.5245	2.5926	**2.6678**
	80%	4.9713	5.3007	**5.4765**	4.9022	5.0085

Table 5. Image restoration outputs for NSDY, DY, DDY1, JYJLL, and IDY in terms of PSNR.

METHOD		NSDY	DY	DDY1	JYJLL	IDY
IMAGE	NOISE	PSNR	PSNR	PSNR	PSNR	PSNR
CAMERA	30%	30.6398	**30.9961**	30.5249	30.8717	30.8786
	50%	27.6515	27.2421	27.5891	**27.7729**	27.3638
	80%	23.7462	23.2416	23.3517	23.6676	**23.9367**
BUILDING	30%	29.8875	29.8283	29.7804	**29.9679**	29.6387
	50%	26.4309	26.4154	**26.5512**	26.3220	26.1593
	80%	22.2909	21.9993	22.0771	22.2824	**22.5207**

Figure 4. Camera image corrupted by (a) 30 %, (b) 50% and (c) 80 % salt-and-pepper noise; the restored images using NSDY: (d,e,f), DY: (g,h,i), DDY1: (j,k,l), JYJLL (m,n,o), IDY (p,q,r).

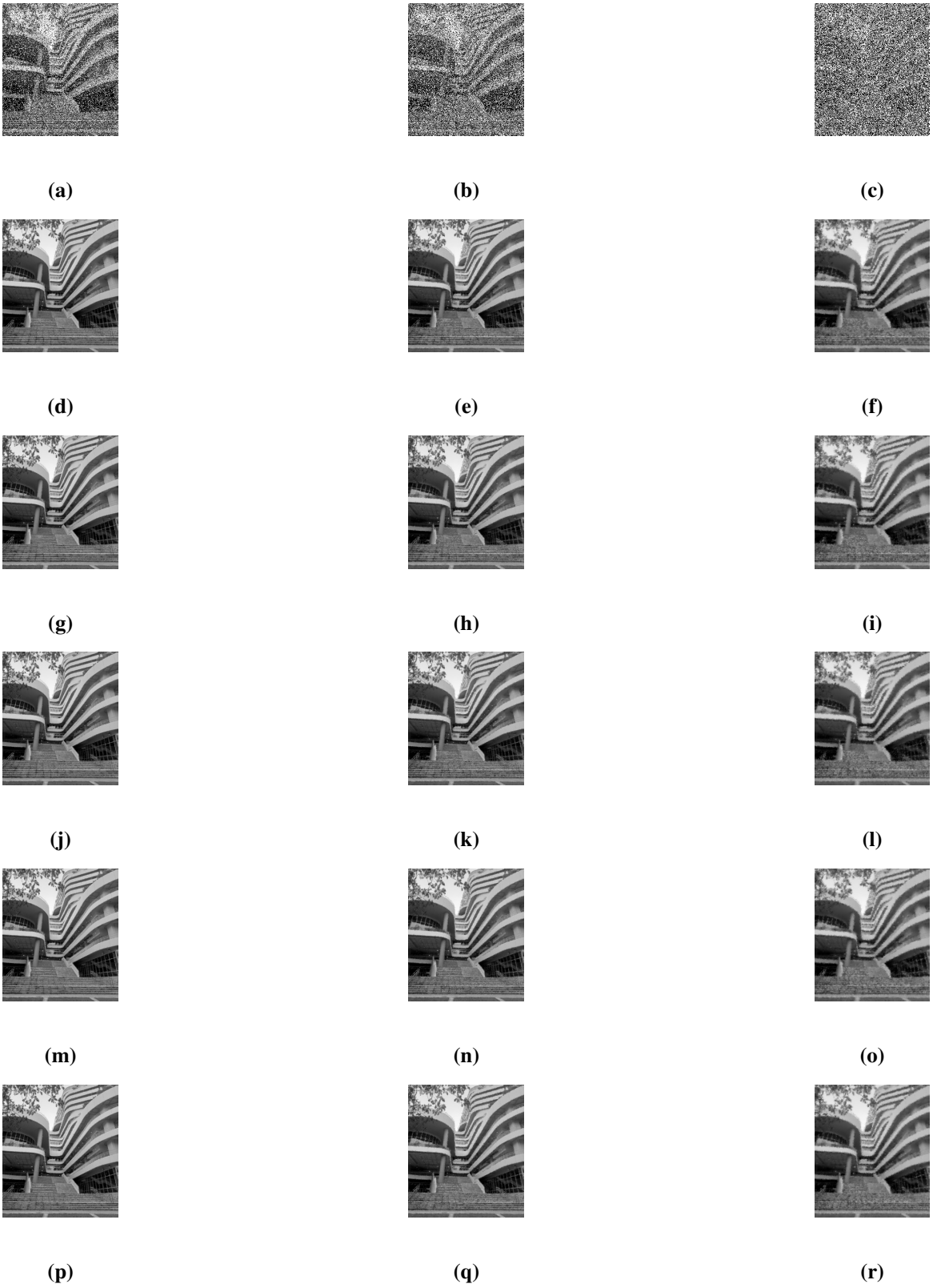

Figure 5. Building image corrupted (a) 30 %, (b) 50% and (c) 80 % salt-and-pepper noise; the restored images using NSDY: (d,e,f), DY: (g,h,i), DDY1: (j,k,l), JYJLL (m,n,o), IDY (p,q,r).

6. Conclusions

In this study, considering the D-L conjugacy condition as well as the Newton search direction, a spectral DY CG method for large-scale unconstrained optimization and image restoration problems is suggested. The proposed method introduces a new spectral parameter θ_k to scale the search direction d_k such that it satisfies the sufficient descent property and converges globally with strong Wolfe criteria. The preliminary computational results a set of optimization functions analyzed in comparison to those obtained for some DY modifications in literature indicated that the proposed method is efficient and reliable.

Use of AI tools declaration

The authors declare that they have not used artificial intelligence tools in the creation of this article.

Acknowledgments

The authors acknowledge the financial support provided by the Center of Excellence in Theoretical and Computational Science (TaCS-CoE), KMUTT and Petchra Pra Jom Klao PhD Scholarship of King Mongkut's University of Technology Thonburi (KMUTT), contract No. 52/2564. Moreover, the authors also acknowledged the financial support provided by the Mid-Career Research Grant (N41A640089).

Conflict of interest

The authors declare no competing interests.

References

1. I. M. Sulaiman, M. Malik, A. M. Awwal, P. Kumam, M. Mamat, S. Al-Ahmad, On three-term conjugate gradient method for optimization problems with applications on covid-19 model and robotic motion control, *Adv. Contin. Discrete Models*, (2022), 1–22.
2. N. Salihu, P. Kumam, A. M. Awwal, I. Arzuka, T. Seangwattana, A Structured Fletcher-Revees Spectral Conjugate Gradient Method for Unconstrained Optimization with Application in Robotic Model, In *Operations Research Forum*, **4** (2023), 81.
3. K. Kamilu, M. Sulaiman, A. Muhammad, A. Mohamad, M. Mamat, Performance evaluation of a novel conjugate gradient method for training feed forward neural network, *Math. Model. Comp.*, **10** (2023), 326–337.
4. M. M. Yahaya, P. Kumam, A. M. Awwal, P. Chaipunya, S. Aji, S. Salisu, A new generalized quasi-newton algorithm based on structured diagonal Hessian approximation for solving nonlinear least-squares problems with application to 3dof planar robot arm manipulator, *IEEE Access*, **10** (2022), 10816–10826. https://doi.org/10.1109/ACCESS.2022.3144875

5. A. S. Halilu, A. Majumder, M. Y. Waziri, K. Ahmed, Signal recovery with convex constrained nonlinear monotone equations through conjugate gradient hybrid approach, *Math. Comput. Simul.*, **187** (2021), 520–539.
6. I. M. Sulaiman, M. Mamat, A new conjugate gradient method with descent properties and its application to regression analysis, *JNAIAM. J. Numer. Anal. Ind. Appl. Math.*, **14** (2020), 25–39.
7. G. Yuan, J. Lu, Z. Wang, The PRP conjugate gradient algorithm with a modified WWP line search and its application in the image restoration problems, *Appl. Numer. Math.*, **152** (2020), 1–11.
8. N. Salihu, P. Kumam, A. M. Awwal, I. M. Sulaiman, T. Seangwattana, The global convergence of spectral RMIL conjugate gradient method for unconstrained optimization with applications to robotic model and image recovery, *Plos one*, **18** (3), e0281250.
9. M. Malik, I. M. Sulaiman, A. B. Abubakar, G. Ardaneswari, Sukono, A new family of hybrid three-term conjugate gradient method for unconstrained optimization with application to image restoration and portfolio selection, *AIMS Math.*, **8** (2023), 1–28.
10. N. Andrei, A Dai-Liao conjugate gradient algorithm with clustering of eigenvalues, *Numer. Algorithms*, **77** (4), 1273–1282.
11. W. W. Hager, H. Zhang, A survey of nonlinear conjugate gradient methods, *Pac. J. Optim.*, **2** (2006), 35–58.
12. R. Fletcher, C. M. Reeves, Function minimization by conjugate gradients, *Comput. J.*, **7** (1964), 149–154. https://doi.org/10.1093/comjnl/7.2.149
13. Y. H. Dai, Y. Yuan, A nonlinear conjugate gradient method with a strong global convergence property, *SIAM J. Optim.*, **10** (1999), 177–182. https://doi.org/10.1137/S1052623497318992
14. R. Fletcher, *Practical methods of optimization*, A Wiley-Interscience Publication. John Wiley & Sons, Ltd., Chichester, second edition, 1987.
15. X. Du, P. Zhang, W. Ma, Some modified conjugate gradient methods for unconstrained optimization, *J. Comput. Appl. Math.*, **305** (2016), 92–114.
16. M. J. D. Powell, Restart procedures for the conjugate gradient method, *Math. Program.*, **12** (1997), 241–254. https://doi.org/10.1023/A:1007963324520
17. G. Zoutendijk, Nonlinear programming, computational methods, *In: J. Abadie Ed., Integer and Nonlinear Programming, North-Holland, Amsterdam*, 37–86, 1970.
18. N. Andrei, An adaptive scaled BFGS method for unconstrained optimization, *Numer. Algorithms*, **77** (2018), 413–432. https://doi.org/10.1007/s11075-017-0321-1
19. Y. H. Dai, L. Z. Liao, New conjugacy conditions and related nonlinear conjugate gradient methods, *Appl. Math. Optim.*, **43** (2001), 87–101.
20. M. Raydan, The Barzilai and Borwein gradient method for the large scale unconstrained minimization problem, *SIAM J. Optim.*, **7** (1997), 26–33. https://doi.org/10.1137/S1052623494266365
21. J. Barzilai, J. M. Borwein, Two-point step size gradient methods. *IMA J. Numer. Anal.*, **8** (1988), 141–148. https://doi.org/10.1093/imanum/8.1.141

22. E. G. Birgin, J. M. Martínez, A spectral conjugate gradient method for unconstrained optimization, *Appl. Math. Optim.*, **43** (2001), 117–128.

23. N. Salihu, M. R. Odekunle, A. M. Saleh, S. Salihu. A Dai-Liao hybrid Hestenes-Stiefel and Fletcher-Revees methods for unconstrained optimization, *Int. J. Indu. Optim.*, **2** (2021), 33–50.

24. N. Salihu, M. Odekunle, M. Waziri, A. Halilu, A new hybrid conjugate gradient method based on secant equation for solving large scale unconstrained optimization problems, *Iran. J. Optim.*, **12** (2020), 33–44. https://doi.org/10.11606/issn.1984-5057.v12i2p33-44

25. S. Nasiru, R. O. Mathew, Y. W. Mohammed, S. H. Abubakar, S. Suraj, A Dai-Liao hybrid conjugate gradient method for unconstrained optimization, *Int. J. Indu. Optim.*, **2** (2021), 69–84.

26. T. Barz, S. Körkel, G. Wozny, Nonlinear ill-posed problem analysis in model-based parameter estimation and experimental design, *Compu. Chem. Engi.*, **77** (2015), 24–42.

27. A. M. Awwal, I. M. Sulaiman, M. Maulana, M. Mustafa, K. Poom, S. Kanokwan, A spectral RMIL+ conjugate gradient method for unconstrained optimization with applications in portfolio selection and motion control, *IEEE Access*, **9** (2021), 75398–75414. https://doi.org/10.1109/ACCESS.2021.3081570

28. H. Shao, H. Guo, X. Wu, P. Liu, Two families of self-adjusting spectral hybrid DL conjugate gradient methods and applications in image denoising, *Appl. Math. Model.*, **118** (2023), 393–411.

29. J. Jian, P. Liu, X. Jiang, C. Zhang, Two classes of spectral conjugate gradient methods for unconstrained optimizations, *J. Appl. Math. Comput.*, **68** (2022), 4435–4456. https://doi.org/10.1007/s12190-022-01713-2

30. J. Jian, P. Liu, X. Jiang, B. He, Two improved nonlinear conjugate gradient methods with the strong Wolfe line search, *Bull. Iranian Math. Soc.*, **48** (2022), 2297–2319. https://doi.org/10.1007/s41980-021-00647-y

31. I. Arzuka, M. R. Abu Bakar, W. J. Leong, A scaled three-term conjugate gradient method for unconstrained optimization, *J. Ineq. Appl.*, (2016), 1–16. https://doi.org/10.15600/2238-1244/sr.v16n42p1-10

32. X. Jiang, J. Jian, Improved Fletcher-Reeves and Dai-Yuan conjugate gradient methods with the strong Wolfe line search, *J. Comput. Appl. Math.*, **34** (2019), 525–534.

33. J. Jian, L. Yang, X. Jiang, P. Liu, M. Liu, A spectral conjugate gradient method with descent property, *Mathematics*, **8** (2020), 280.

34. J. Jian, L. Han, X. Jiang, A hybrid conjugate gradient method with descent property for unconstrained optimization, *Appl. Math. Model.*, **39** (2015), 1281–1290. https://doi.org/10.1016/j.apm.2014.08.008

35. X. Zhou, L. Lu, The global convergence of modified DY conjugate gradient methods under the wolfe line search, *J. Chongqing Normal Univ.(Nat. Sci. Ed.)*, **33** (2016), 6–10.

36. Z. Zhu, D. Zhang, S. Wang, Two modified DY conjugate gradient methods for unconstrained optimization problems, *Appl. Math. Comput.*, **373** (2020), 125004. https://doi.org/10.1016/j.amc.2019.125004

37. N. Andrei, *Nonlinear conjugate gradient methods for unconstrained optimization*, volume 158 of *Springer Optimization and Its Applications*, Springer, Cham, 2020.

38. J. Momin, Y. X. She, A literature survey of benchmark functions for global optimization problems, *Int. J. Mathe. Model. Nume. Optim.*, **4** (2013), 150–194.

39. E. D. Dolan, J. J. Moré, Benchmarking optimization software with performance profiles, *Math. Program.*, **91** (2002), 201–213. https://doi.org/10.1007/s101070100263

40. M. Nadipally, *Chapter 2-Optimization of methods for image-texture segmentation using ant colony optimization*, volume 1, In: Intelligent Data Analysis for Biomedical Applications, Academic Press, Elsevier, 2019.

41. G. Yu, J. Huang, Y. Zhou, A descent spectral conjugate gradient method for impulse noise removal, *Appl. Math. Lett.*, **23** (2010), 555–560. https://doi.org/10.1016/S0268-005X(08)00209-9

AIMS Press