Hindawi
Mathematical Problems in Engineering
Volume 2020, Article ID 6279543, 13 pages
https://doi.org/10.1155/2020/6279543

Research Article

A Modified Dai–Liao Conjugate Gradient Method with a New Parameter for Solving Image Restoration Problems

Junyu Lu,[1] Yong Li ,[2] and Hongtruong Pham[3]

[1]*College of Mathematics and Information Science, Guangxi Center for Mathematical Research, Guangxi University, Nanning, Guangxi, China*
[2]*College of Mathematics and Statistics, Baise University, Baise, China*
[3]*Thai Nguyen University of Economics and Business Administration, Thai Nguyen, Vietnam*

Correspondence should be addressed to Yong Li; liyong3922@bsuc.edu.cn

Received 25 July 2020; Revised 22 August 2020; Accepted 29 August 2020; Published 18 September 2020

Academic Editor: Maojun Zhang

One adaptive choice for the parameter of the Dai–Liao conjugate gradient method is suggested in this paper, which is obtained with modified quasi–Newton equation. So we get a modified Dai–Liao conjugate gradient method. Some interesting features of the proposed method are introduced: (i) The value of parameter t of the modified Dai–Liao conjugate gradient method takes both the gradient and function value information. (ii) We establish the global convergence property of the modified Dai–Liao conjugate gradient method under some suitable assumptions. (iii) Numerical results show that the modified DL method is effective in practical computation and the image restoration problems.

1. Introduction

In this paper, we consider the following unconstrained optimization problem:

$$\min f(x), \quad x \in \mathfrak{R}^n, \tag{1}$$

where $f\colon \mathfrak{R}^n \longrightarrow \mathfrak{R}$ and f is a continuously differentiable function. Requiring simplicity and low memory storage [1–5], conjugate gradient (CG) methods [6–11] are useful tools for solving large-scale problems. Thus, we consider solving (1) with the conjugate gradient methods. A sequence of iterates is generated from a given x_0 with the following simple iteration formula:

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

where x_k is the k-th iteration point, the step size $\alpha_k > 0$ can be usually chosen to satisfy certain line search conditions, and d_k is the search direction defined by

$$d_k = \begin{cases} -g_k, & \text{if } k = 0, \\ -g_k + \beta_k d_{k-1}, & \text{if } k \geq 1, \end{cases} \tag{3}$$

where β_k is a scalar. Many scholars presented some classical formulas of β_k as follows: the corresponding methods are called the Polak–Ribiére–Polyak (PRP) [12, 13], Fletcher–Reeves (FR) [14], Hestense–Stiefel (HS) [15], conjugate descent (CD) [16], Liu–Storey (LS) [17], and Dai–Yuan (DY) [18] CG methods:

$$\beta_k^{\mathrm{PRP}} = \frac{g_k^T (g_k - g_{k-1})}{\|g_{k-1}\|^2},$$

$$\beta_k^{\mathrm{FR}} = \frac{\|g_k\|^2}{\|g_{k-1}\|^2},$$

$$\beta_k^{\mathrm{HS}} = \frac{g_k^T (g_k - g_{k-1})}{d_{k-1}^T (g_k - g_{k-1})},$$

$$\beta_k^{\mathrm{CD}} = \frac{\|g_k\|^2}{-d_{k-1}^T g_{k-1}},$$

$$\beta_k^{\mathrm{LS}} = \frac{g_k^T (g_k - g_{k-1})}{-d_{k-1}^T g_{k-1}},$$
$$\beta_k^{\mathrm{DY}} = \frac{\|g_k\|^2}{(g_k - g_{k-1})^T d_{k-1}}, \tag{4}$$

where g_{k-1} is the gradient $\nabla f(x_{k-1})$ of $f(x)$ at the point x_{k-1} and $\|\cdot\|$ is the Euclidean norm. We relatively easily established the global convergence property of the CD method, DY method, and FR method, but numerical results of these methods are not desirable in some computations. Powell [19] explained a major numerical disadvantage of the FR method, such as subsequent steps being very short if a small step is originated away from the solution point. If poor d_k occurs in practical computation, the PRP, HS, or LS method will perform a restart, so these three methods perform much better than the above three methods in the numerical test. They are generally regarded as the most efficient conjugate gradient methods. Therefore, in recent year, many scholars try to research some modified formulas for conjugate gradient methods which have global convergence property for general functions and satisfied performance in the numerical test. Among them, Dai and Liao [20] proposed some modified conjugate methods with a new conjugacy condition. Interestingly, their method cited in [20] not only has global convergence for general function, but also has better numerical performance than HS and PR methods.

When the CG methods produce a sequence of search directions d_k, the following conjugacy condition holds:

$$d_m^T H d_n = 0, \tag{5}$$

where $m \neq n$ and H is the Hessian of the objective function. The definition of y_{k-1} is as follows:

$$y_{k-1} = g_k - g_{k-1}. \tag{6}$$

If the general function f is nonlinear, combining with the knowledge of mean value theorem, we draw a conclusion such that

$$d_k^T y_k = \alpha_{k-1} d_k^T \nabla^2 f(x_{k-1} + t\alpha_{k-1} d_{k-1}) d_{k-1}, \tag{7}$$

where $t \in (0, 1)$. By (5) and (7), we have

$$d_k^T y_{k-1} = 0. \tag{8}$$

As we know, Dai and Liao [20] studied (8) in depth based on the quasi–Newton techniques. We have that in the quasi–Newton method, H_{k-1} is defined as an approximation matrix of the Hessian $\nabla^2 f(x_{k-1})$, and the new matrix H_k holds the following formula:

$$H_k s_{k-1} = y_{k-1}. \tag{9}$$

In the quasi–Newton method, the search direction d_k is defined by

$$d_k = -H_k^{-1} g_k. \tag{10}$$

According to the above two equations, we have

$$d_k^T y_{k-1} = d_k^T (H_k s_{k-1}) = -g_k^T s_{k-1}. \tag{11}$$

Based on the above relations, Dai and Liao acquired the following conjugacy condition:

$$d_k^T y_{k-1} = -t g_k^T s_{k-1}, \tag{12}$$

where t is a positive parameter. In the case $t = 0$, (12) becomes (8). In the case $t = 1$, (12) becomes (11). Moreover, $g_k^T s_{k-1} = 0$ holds under the exact line search. It is easily observed that both (11) and (12) coincide with (8). According to the above discussion, (12) can be considered as an extension of (8) and (11). Multiplying the definition of d_k with y_{k-1} and using (12), Dai and Liao introduced the new formula in the DL method as follows:

$$\beta_k^{\mathrm{DL}} = \frac{g_k^T y_{k-1} - t g_k^T s_{k-1}}{d_{k-1}^T y_{k-1}}. \tag{13}$$

In [20], the conjugate gradient method with (13) establishes the global convergence property of uniformly convex functions. Furthermore, in order to ensure general functions also satisfy global convergence, a new formula was presented by Dai and Liao as follows:

$$\beta_k^{\mathrm{DL+}} = \max\left\{ \frac{g_k^T y_{k-1}}{d_{k-1}^T y_{k-1}}, 0 \right\} - \frac{t g_k^T s_{k-1}}{d_{k-1}^T y_{k-1}}. \tag{14}$$

It is easily observed that the values of the first term in (13) are nonnegative. They proved that the modified DL method with (14) satisfies global convergence for general functions under some suitable conditions. In the DL method, different parameter t shows different numerical performance. Based on the singular value study, Hager and Zhang [21] presented the new choices for t:

$$t = \frac{2\|y_k\|^2}{s_k^T y_k}, \tag{15}$$

and they proved that d_k in [21] satisfies the sufficient descent condition. In addition, Babaie-Kafaki and Ghanbari [22] presented a new value for t:

$$t_1 = \frac{\|s_k\|^2 \|y_k\|^2}{(s_k^T y_k)^2}. \tag{16}$$

In [22], they proved that the DL method with $t = t_1$ has better numerical results than the methods proposed by Dai and Kou [23]. In last several years, some scholars tried to find the new choice for the nonnegative parameter t in (13) [24, 25]. The remainder of this paper is organized as follows: in Section 2, based on the new conjugacy condition, a modified DL gradient method is proposed with a new value for the parameter. We establish the global convergence property of the presented method in Section 3. Finally, we do some numerical experiments in Section 4.

2. New DL Conjugate Gradient Method

It is well known that the objective function can be regarded as a quadratic equation, so, if a point x_k is close enough to a local minimizer, a good direction is the Newton direction:

$$d_k = -\nabla^2 f(x_k)^{-1} g_k. \tag{17}$$

Hence, from d_k of CG methods and the formula of β_k in (13), we can compute the value of parameter t as follows:

$$-\nabla^2 f(x_k)^{-1} g_k = -g_k + \frac{g_k^T y_{k-1}}{d_{k-1}^T y_{k-1}} d_{k-1} - \frac{t g_k^T s_{k-1}}{d_{k-1}^T y_{k-1}} d_{k-1}. \tag{18}$$

By some algebra, we obtain

$$t = \frac{s_{k-1}^T g_k - s_{k-1}^T \nabla^2 f(x_k) g_k + \left(g_k^T y_{k-1}/s_{k-1}^T y_{k-1}\right) s_{k-1}^T \nabla^2 f(x_k) s_{k-1}}{\left(g_k^T s_{k-1}/s_{k-1}^T y_{k-1}\right) s_{k-1}^T \nabla^2 f(x_k) s_{k-1}}. \tag{19}$$

Thoughtfully, because computing the Hessian matrix $\nabla^2 f(x_k)$ is inefficient, a quasi–Newton method is used in this paper. In the quasi–Newton methods [26], the search direction d_k is computed by solving unconstrained optimization problems, $B_k d_k = -g_k$, where B_k is the approximation matrix of the Hessian $\nabla^2 f(x_k)$ such that the matrix B_k satisfies the following important equation:

$$B_k s_{k-1} = y_{k-1}. \tag{20}$$

Theorem 1. *Assume that the function $f(x)$ is sufficiently smooth and $\|s_{k-1}\|$ is sufficiently small, and then, we have*

$$s_{k-1}^T \nabla^2 f(x_k) s_{k-1} - s_{k-1}^T y_{k-1} = \frac{1}{2} s_{k-1}^T (T_k s_{k-1}) s_{k-1} + O\left(\|s_{k-1}\|^4\right), \tag{21}$$

where T_k is the tensor of f at x_k and

$$s_{k-1}^T (T_k s_{k-1}) s_{k-1} = \sum_{i,j,l=1}^{n} \frac{\partial^3 f(x_k)}{\partial x^i \partial x^j \partial x^l} s_k^i s_k^j s_k^l. \tag{22}$$

Proof. Doing Taylor expansion for the objective function $f(x)$, we have

$$\begin{aligned} f_{k-1} &= f_k - g_k^T s_{k-1} + \frac{1}{2} s_{k-1}^T \nabla^2 f(x_k) s_{k-1} \\ &\quad - \frac{1}{6} s_{k-1}^T (T_k s_{k-1}) s_{k-1} + O\left(\|s_{k-1}\|^4\right), \\ g_{k-1}^T s_{k-1} &= g_k^T s_{k-1} - s_{k-1}^T \nabla^2 f(x_k) s_{k-1} + \frac{1}{2} s_{k-1}^T (T_k s_{k-1}) s_{k-1} \\ &\quad + O\left(\|s_{k-1}\|^4\right). \end{aligned} \tag{23}$$

By the definitions of y_{k-1}, we completed the proof.

The BFGS-type methods have been proven to satisfy ideal global convergence of uniformly convex functions, but the methods may fail to establish the convergence property of nonconvex functions. So, a new version of the BFGS method is proposed by Wei et al. [27] to overcome its convergence failure for more general objective function. In their method, the matrix B_k is obtained from the following modified secant equation:

$$\begin{aligned} B_k s_{k-1} &= y_{k-1}^*, \\ y_{k-1}^* &= y_{k-1} + \nu_{k-1} s_{k-1}, \end{aligned} \tag{24}$$

where $\nu_{k-1} = (\max(\vartheta_{k-1}, 0)/\|s_{k-1}\|^2)$ and $\vartheta = 2(f_{k-1} - f_k) + (g_{k-1} + g_k)^T s_{k-1}$. In order to get the powerful theoretical and numerical properties of the modified BFGS method, use (24) to simplify (19) and the parameter t is proposed as follows:

$$t_2 = \frac{(1 - \nu_{k-1}) s_{k-1}^T g_k + \left(g_k^T y_{k-1}/s_{k-1}^T y_{k-1}\right) \nu_{k-1} \|s_{k-1}\|^2}{g_k^T s_{k-1} + \left(g_k^T s_{k-1}/s_{k-1}^T y_{k-1}\right) \nu_{k-1} \|s_{k-1}\|^2}. \tag{25}$$

Then, to ensure that the new DL method satisfies the descent condition, we present a modified definition of the value of parameter t similar to [28] as follows:

$$t^* = \max\left(t_2, \theta \frac{\|y_{k-1}\|^2}{s_{k-1}^T y_{k-1}}\right), \tag{26}$$

where $\theta > (1/4)$; it means that we have d_k as follows:

$$d_k = -g_k + \beta_k^{\mathrm{DL}*} d_{k-1}, \tag{27}$$

where $\beta_k^{\mathrm{DL}*} = ((g_k^T y_{k-1} - t^* g_k^T s_{k-1})/d_{k-1}^T y_{k-1})$. Based on the above discussion, we can collect some of the advantages of these formulas:

(i) The new DL method possesses the information about function values.

(ii) The global convergence for nonconvex functions is established under some assumptions.

(iii) The new formulas are applied to the engineering Muskingum model and image restoration problems. Finally, we end this section by presenting a modified DL algorithm, which is listed in Algorithm 1. □

3. Convergence Analysis

In the following paper, there are some indispensable assumptions for the global convergence of the algorithm on objective functions.

Assumption 1. (i) The level set $Y = \{x \mid f(x) \leq f(x_0)\}$ is bounded.

(ii) In some neighbourhood Ψ of Υ, f is differentiable, and its gradient function g is the Lipschitz continuous, namely,

Step 0: choose an initial point $x_0 \in \mathfrak{R}^n$, $\delta \in (0,1)$, $\delta_1 \in (0,\delta)$, $\sigma \in (\delta,1)$, $\varepsilon > 0$. Set $d_0 = -g_0 = -\nabla f(x_0)$, $k := 0$.
Step 1: stop if $\|g_k\| \le \varepsilon$.
Step 2: determine a stepsize satisfying the following line search by [4]: $f(x_k + \alpha_k d_k) \le f_k + \delta\alpha_k g_k^T d_k + \alpha_k \min[-\delta_1 g_k^T d_k, \delta(\alpha_k/2)\|d_k\|^2]$, and $g(x_k + \alpha_k d_k)^T d_k \ge \sigma g_k^T d_k + \min[-\delta_1 g_k^T d_k, \delta\alpha_k\|d_k\|^2]$
where $\delta \in (0, (1/2))$, $\delta_1 \in (0,\delta)$, and $\sigma \in (\delta, 1)$.
Step 3: let $x_{k+1} = x_k + \alpha_k d_k$.
Step 4: if $\|g_{k+1}\| \le \varepsilon$, then the modified DL algorithm stops.
Step 5: calculate the search direction by (27).
Step 6: set $k := k+1$ and go to step 3.

Algorithm 1: YWLDL algorithm.

$$\|g(x) - g(y)\| \le L\|x - y\|, \tag{28}$$

where $L > 0$ is a constant and any $x, y \in \Psi$.

Remark 1. Assumption 1 implies that there exists a constant $\eta > 0$ satisfying

$$\|g(x)\| \le \eta, \quad \forall x \in \Psi. \tag{29}$$

Remark 2. In order to establish the convergence property of new method, we assume the new proposed parameter is bounded. For this purpose, we set

$$t^* \le H, \tag{30}$$

where H is a large positive constant.

Lemma 1. *Consider the proposed modified DL method in the form of (27). If the line search procedure guarantees $d_k^T y_k \ge 0$, for all $k \ge 0$, then, we have*

$$g_k^T d_k \le -\lambda\|g_k\|^2, \tag{31}$$

where $\lambda = (1 - (1/4\theta))$.

Proof. Similar to the proof of Theorem 1 [29]. □

Lemma 2. *Suppose that Assumption 1 holds. Consider the proposed modified DL method in the form of (27), and the step size α_k obtained by the equations in step 2 in Algorithm 1. If f be a nonconvex function on the level set Ψ, then, we have*

$$\sum_{k=0} \frac{(g_k^T d_k)^2}{\|d_k\|^2} < \infty. \tag{32}$$

Proof. From Lemma 1, $d_k^T g_k < 0$ holds. By the second equation in Step 2 in Algorithm 1 and (28),

$$\begin{aligned} \alpha_k L\|d_k\|^2 &\ge (g_{k+1} - g_k)^T d_k \ge (\sigma - 1) g_k^T d_k \\ &\quad + \min\left[-\delta_1 g_k^T d_k, \delta\alpha_k\|d_k\|^2\right] \ge (\sigma - 1) g_k^T d_k. \end{aligned} \tag{33}$$

So the following equation holds:

$$\alpha_k \ge \frac{(\sigma - 1) g_k^T d_k}{L\|d_k\|^2}. \tag{34}$$

From the first equation in Step 2 in Algorithm 1, we obtain

$$\begin{aligned} f(x_k) - f(x_k + \alpha_k d_k) &\ge -\delta\alpha_k g_k^T d_k \\ &\quad - \alpha_k \min\left[-\delta_1 g_k^T d_k, \delta\frac{\alpha_k}{2}\|d_k\|^2\right] \ge -(\delta - \delta_1)\alpha_k g_k^T d_k. \end{aligned} \tag{35}$$

Summing up both sides of the above inequalities from $k = 0$ to ∞, we obtain

$$\sum_{k=0}^{\infty} \left(-\alpha_k g_k^T d_k\right) < \infty. \tag{36}$$

Combining with (34), (32) holds, which completes the proof. □

Lemma 3. *Suppose that Assumption 1 is true and then the sequence $\{x_k, d_k, \alpha_k, g_k\}$ is generated by Algorithm YWLDL. If*

$$\|g_k\| \ge \tau, \tag{37}$$

we obtain

$$\sum_{k=0}^{\infty} \frac{1}{\|d_k\|^2} < \infty. \tag{38}$$

Proof. From Lemma 1 and (37), we have

$$\tau^2 \sum_{k=0}^{\infty} \frac{1}{\|d_k\|^2} \le \sum_{k=0}^{\infty} \frac{\|g_k\|^4}{\|d_k\|^2} \le \sum_{k=0}^{\infty} \frac{1}{\lambda^2} \frac{(g_k^T d_k)^2}{\|d_k\|^2}. \tag{39}$$

Combining with Lemma 2, such that

$$\sum_{k=0}^{\infty} \frac{1}{\|d_k\|^2} < \infty. \tag{40}$$

The proof is complete. □

Theorem 2. *Suppose that Assumption 1 holds. Consider that the proposed modified DL method in the form of (27) and the step size α_k obtained the equations in step 2 in Algorithm 1. If f be a nonconvex function on the level set Ψ, then, we obtain*

$$\lim_{k\longrightarrow\infty} \inf\left\|g_k\right\| = 0. \tag{41}$$

Proof. First, we suppose that equation (41) is not true. Namely, equation (37) holds. From the equations in step 2 in Algorithm 1, in case 1, $-\delta_1 g_k^T d_k \geq \delta\alpha_k\|d_k\|^2$ holds, combining with (36), we have

$$\sum_{k=0}^{\infty}\left\|s_k\right\|^2 = \sum_{k=0}^{\infty}\left\|\alpha_k d_k\right\|^2 \leq \sum_{k=0}^{\infty}\left(-\frac{\delta_1}{\delta}\alpha_k g_k^T d_k\right) < \infty. \tag{42}$$

And in case 2, similar to the proof of Theorem 3.1 [30], the relation holds as follows:

$$\alpha_k\left\|d_k\right\|^2 \leq -\overline{\delta} g_k^T d_k, \tag{43}$$

where $\overline{\delta} > 0$. From (36) and (43), the following equation holds in case 2:

$$\sum_{k=0}^{\infty}\left\|s_k\right\|^2 \leq c\sum_{k=0}^{\infty}\left\|\alpha_k d_k\right\|^2 \leq c\sum_{k=0}^{\infty}\left(-\frac{1}{\overline{\delta}}\alpha_k g_k^T d_k\right) < \infty. \tag{44}$$

Thus,

$$\left\|s_k\right\| \longrightarrow 0, \quad \forall k > 0. \tag{45}$$

We have

$$\left\|s_k\right\| < z, \tag{46}$$

where $0 < z < 1$ holds. And from the second equation in Step 2 in Algorithm 1, we obtain

$$\begin{aligned} y_k^T d_k &= g_{k+1}^T d_k - g_k^T d_k \geq (\sigma - 1) g_k^T d_k \\ &\quad + \min\left[-\delta_1 g_k^T d_k, \delta\alpha_k\left\|d_k\right\|^2\right] \geq (\sigma - 1) g_k^T d_k. \end{aligned} \tag{47}$$

By (31) and (37), we have

$$y_k^T d_k \geq (1 - \sigma)\lambda\tau^2. \tag{48}$$

Combining with (30)–(30), (46), and (48), the following equation holds:

$$\begin{aligned} \left|\beta_k^{\mathrm{DL}*}\right| &= \left|\frac{g_k^T y_{k-1} - t^* g_k^T s_{k-1}}{d_{k-1}^T y_{k-1}}\right| \leq \frac{(L+H)\left\|g_k\right\|\left\|s_{k-1}\right\|}{d_{k-1}^T y_{k-1}} \\ &\leq \frac{(L+H)\eta z}{d_{k-1}^T y_{k-1}} \leq \frac{(L+H)\eta z}{(1-\sigma)\lambda\tau^2} = Z, \end{aligned} \tag{49}$$

where $Z = ((L+H)\eta z/(1-\sigma)\lambda\tau^2)$. From (27), we have

$$\begin{aligned} \left\|d_k\right\| &= \left\|-g_k + \beta_k^{\mathrm{DL}*} d_{k-1}\right\| \leq \eta + Z\left\|d_{k-1}\right\| \\ &\leq \eta\left[1 + Z + (Z)^2 + \cdots + (Z)^{k-k_0-1}\right] \\ &\quad + (Z)^{k-k_0}\left\|d_{k_0}\right\| \leq \frac{\eta}{1-Z} + \left\|d_{k_0}\right\|. \end{aligned} \tag{50}$$

Let $v = \max\left\{\|d_1\|, \|d_2\|, \ldots, \|d_{k_0}\|, (\eta/(1-Z)) + \|d_{k_0}\|\right\}$, the following holds:

$$\left\|d(x_k)\right\| \leq v, \quad \forall k \geq 0. \tag{51}$$

This contradicts (38), so the proof is completed. □

4. Numerical Experiments

This section reports some numerical results. We do some numerical experiments in order to investigate the performance of our proposed algorithm and compare it with others. In the next subsections, all the numerical tests were run in a 2.30 GHz CPU and 8.00 GB of memory in a Windows 10 operating system.

4.1. Normal Unconstrained Optimization Problems. In this section, we test the effect of some algorithms on numerical problems, and the unconstrained optimization problems are listed in Table 1. We will report on various numerical implementations of our presented method (YWLDL algorithm) with the modified Wolfe–Powell line search (YWL) on these problems and compare its performance with the one of the following algorithms: HZ algorithm, the BG algorithm, and the DL+ algorithm, for the problems. The proposed algorithm is listed as follows:

HZ algorithm: the DL method with parameter t in [21] under the weak Wolfe–Powell line search

BG algorithm: the DL method with parameter t in [22] under the weak Wolfe–Powell line search

DL+ algorithm: the DL method with parameter t in [20] under the strong Wolfe–Powell line search

In this part of the numerical experiment, we introduce the stopping rules, dimensions, and some parameters as follows:

Stop rules (the Himmelblau stop rule [31]): if $|f(x_k)| > e_1$, let $\text{stop}1 = (|f(x_k) - f(x_{k+1})|/|f(x_k)|)$ or $\text{stop}1 = |f(x_k) - f(x_{k+1})|$. If the condition $\|g(x)\| < \varepsilon$ or $\text{stop}1 < e_2$ is satisfied, where $e_1 = e_2 = 10^{-6}$, $\varepsilon = 10^{-6}$. The algorithm will stop if the number of iterations is greater than 1000.

Dimension: 3000, 6000, and 9000 variables.

Parameters: $\delta = 0.2$, $\delta_1 = 0.05$, and $\sigma = 0.9$.

The columns of Table 1 have the following meanings:

Nr: the number of tested problems

Test problems: the name of problems

Table 1: Test problems.

Nr.	Test problems
1	Extended Freudenstein and Roth function
2	Extended trigonometric function
3	Extended Rosenbrock function
4	Extended white and holst function
5	Extended beale function
6	Extended penalty function
7	Perturbed quadratic function
8	Raydan 1 function
9	Raydan 2 function
10	Diagonal 1 function
11	Diagonal 2 function
12	Diagonal 3 function
13	Hager function
14	Generalized tridiagonal 1 function
15	Extended tridiagonal 1 function
16	Extended three exponential terms function
17	Generalized tridiagonal 2 function
18	Diagonal 4 function
19	Diagonal 5 function
20	Extended Himmelblau function
21	Generalized PSC1 function
22	Extended PSC1 function
23	Extended Powell function
24	Extended block diagonal BD1 function
25	Extended Maratos function
26	Extended cliff function
27	Quadratic diagonal perturbed function
28	Extended wood function
29	Extended Hiebert function
30	Quadratic function QF1 function
31	Extended quadratic penalty QP1 function
32	Extended quadratic penalty QP2 function
33	A quadratic function QF2 function
34	Extended EP1 function
35	Extended tridiagonal-2 function
36	BDQRTIC function (CUTE)
37	TRIDIA function (CUTE)
38	ARWHEAD function (CUTE)
39	ARWHEAD Function (CUTE)
40	NONDQUAR function (CUTE)
41	DQDRTIC function (CUTE)
42	EG2 function (CUTE)
43	DIXMAANA function (CUTE)
44	DIXMAANB function (CUTE)
45	DIXMAANC function (CUTE)
46	DIXMAANE function (CUTE)
47	Partial perturbed quadratic function
48	Broyden tridiagonal function
49	Almost perturbed quadratic function
50	Tridiagonal perturbed quadratic function
51	EDENSCH function (CUTE)
52	VARDIM function (CUTE)
53	STAIRCASE S1 function
54	LIARWHD function (CUTE)
55	DIAGONAL 6 function
56	DIXON3DQ function (CUTE)
57	DIXMAANF function (CUTE)
58	DIXMAANG function (CUTE)
59	DIXMAANH function (CUTE)
60	DIXMAANI function (CUTE)
61	DIXMAANJ function (CUTE)
62	DIXMAANK function (CUTE)
63	DIXMAANL function (CUTE)
64	DIXMAAND function (CUTE)
65	ENGVAL1 function (CUTE)
66	FLETCHCR function (CUTE)
67	COSINE function (CUTE)
68	Extended DENSCHNB function (CUTE)
69	DENSCHNF function (CUTE)
70	SINQUAD function (CUTE)
71	BIGGSB1 function (CUTE)
72	Partial perturbed quadratic PPQ2 function
73	Scaled quadratic SQ1 function
74	Scaled quadratic SQ2 function

The comparison data can be shown as follows:

NI: the number of iterations

NFG: the total of the function and gradient evaluations

CPU: the CPU time of the algorithm spent in seconds

We used the tool presented by Dolan and Moré [32] in these numerical reports, the new tool to show their performance in order to analyse the efficiency of the YWLDL algorithm, HZ algorithm, BG algorithm, and DL+ algorithm, and Figures 1–3 show the profiles relative to the CPU time, NI, and NFG, respectively. Figure 1 presents that YWLDL successfully solves about 35 percent of the test problems with the CPU time in seconds, HZ solves about 18 percent of the test problems, BG solves about 24 percent of the test problems, and DL+ solves about 6 percent of the test problems. We can conclude that YWLDL is more competitive than the HZ algorithm, BG algorithm, and DL+ algorithm since its performance curves corresponding to the CPU time are best. Figures 2 and 3 show that the robustness of YWLDL algorithm is worse than that of the BG algorithm. The YWLDL algorithm can successfully solve most of the test problems. Altogether, it is clear that the YWLDL algorithm is efficient based on the experimental results and the YWLDL algorithm with modified Wolfe–Powell line search is competitive with the other four algorithms for these test problems.

4.2. Muskingum Model in Engineering Problems. It is well known that some algorithms for solving optimization problems have been reassessed as a major challenge in engineering problems. Many scholars hope to present some effective algorithms to solve these engineering problems. Parameter estimation is one of the important tools for the research of a well-known hydrologic engineering application problem called the nonlinear Muskingum model. The Muskingum model, depending on the water inflow and outflow, is a popular flood routing model defined as follows:

The Muskingum model [33] is defined by

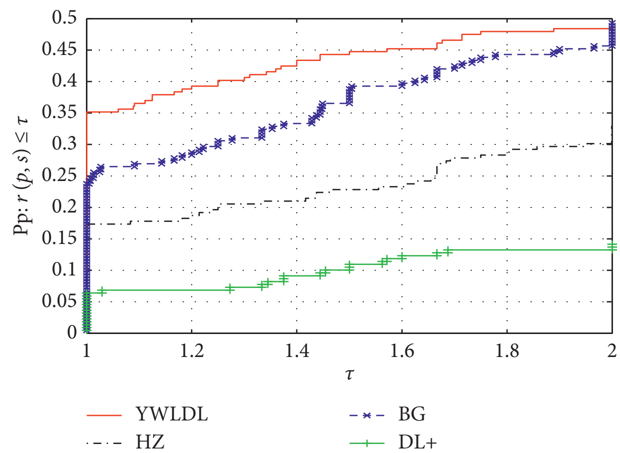

Figure 1: Performance profiles of these methods (CPU).

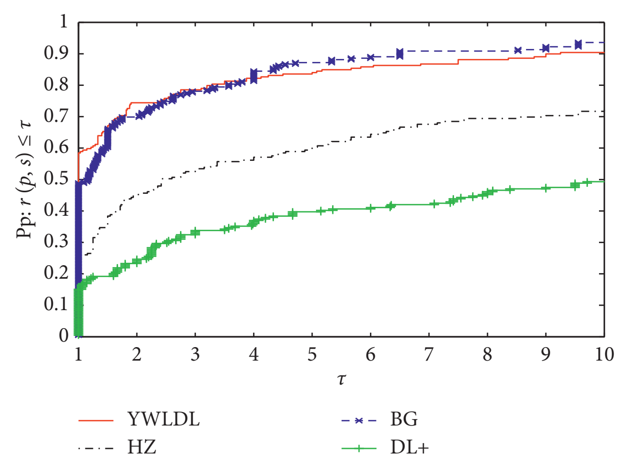

Figure 2: Performance profiles of these methods (NI).

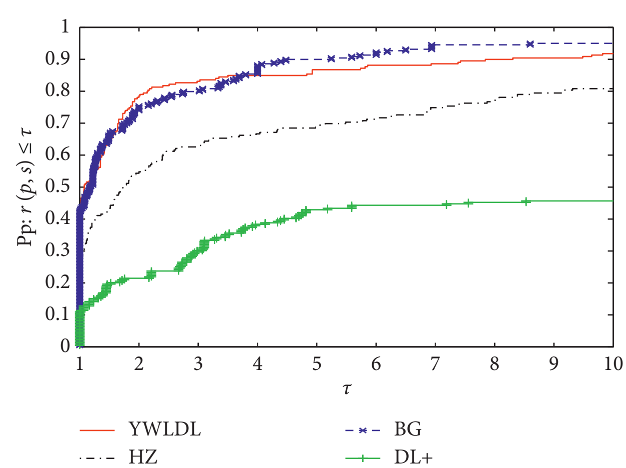

Figure 3: Performance profiles of these methods (NFG).

Table 2: Results of the algorithms.

Algorithm	x_1	x_2	x_3
BFGS [35]	10.8156	0.9826	1.0219
HIWO [33]	13.2813	0.8001	0.9933
Algorithm NDL	11.1937	0.9999	0.9993

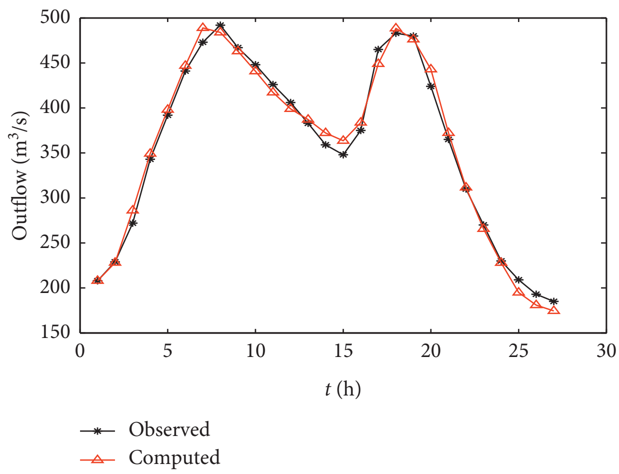

Figure 4: Performance of data in 1960.

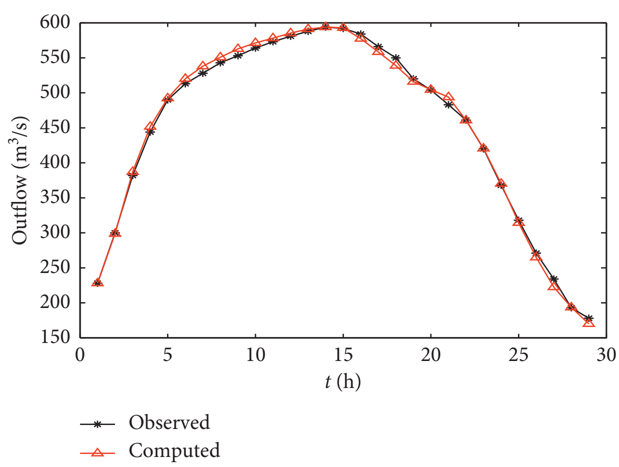

Figure 5: Performance of data in 1961.

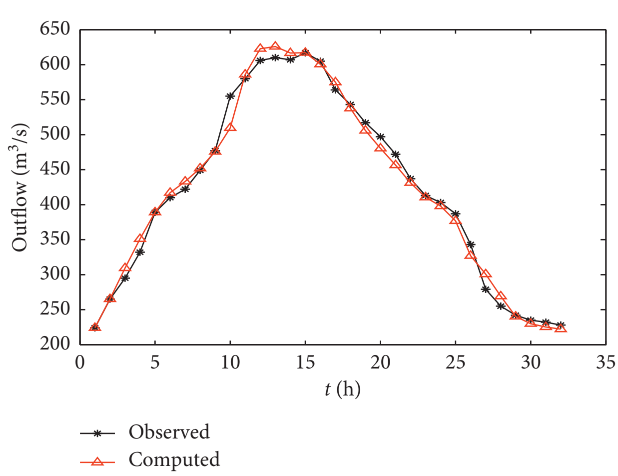

Figure 6: Performance of data in 1964.

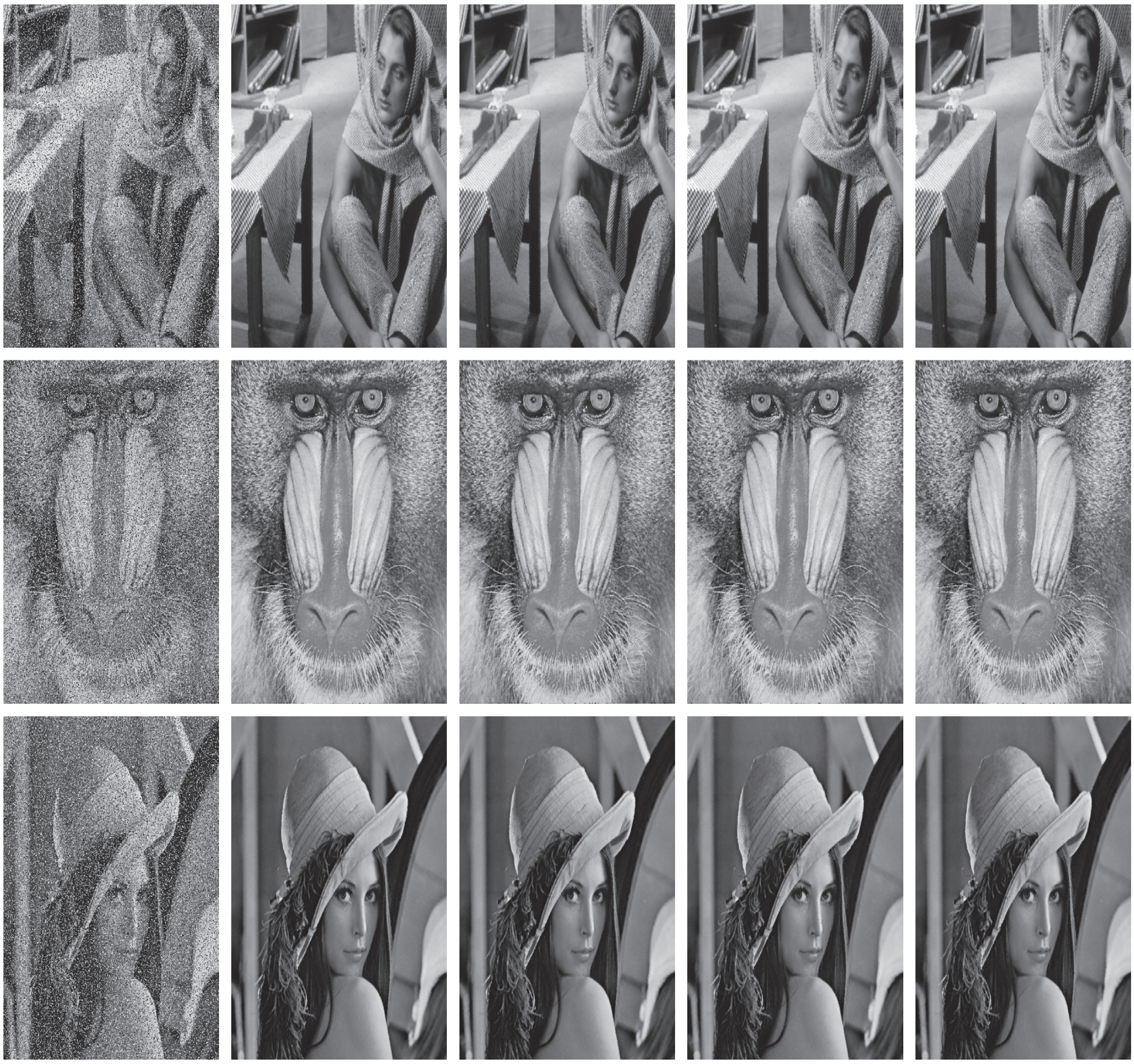

Figure 7: Restoration of the Babara, Baboon, and Lena images by using the HZ algorithm, BG algorithm, DL+ algorithm, and YWLDL algorithm. From left to right: a noisy image with 25% salt-and-pepper noise, restorations obtained by minimizing z with the HZ algorithm, BG algorithm, and DL+ algorithm and YWLDL algorithm.

$$\min f(x_1, x_2, x_3) = \sum_{i=1}^{n-1} \left(\left(1 - \frac{\Delta t}{6}\right) x_1 \left(x_2 I_{i+1} + (1 - x_2) Q_{i+1}\right)^{x_3} \right. \\ - \left(1 - \frac{\Delta t}{6}\right) x_1 \left(x_2 I_i + (1 - x_2) Q_i\right)^{x_3} \\ \left. - \frac{\Delta t}{2} (I_i - Q_i) + \frac{\Delta t}{2} \left(1 - \frac{\Delta t}{3}\right) (I_{i+1} - Q_{i+1}) \right)^2, \tag{52}$$

at time t_k, where $k = 1, 2, \ldots, n$, n denotes the total number of times; x_1, x_2, and x_3 denote the storage time constant; the weighting factor, and an additional parameter, respectively; Δt denotes the time step; I_i denotes the observed inflow discharge; and Q_i denotes the observed outflow discharges. In Chenggouwan and Linqing of Nanyunhe in the Haihe Basin, Tianjin, China, using actual observation data of flood run off process, $\Delta t = 12(h)$ and the initial point $x = [0, 1, 1]^T$ are presented. The detailed data for I_i and Q_i in 1960, 1961, and 1964 can be found in [34]. The tested results of the final points are listed in Table 2.

Figures 4–6 can draw at least three conclusions: (i) It is not different from the BFGS method and HIWO method, the presented algorithm provides a good approximation for these data, and the YWLDL algorithm can effectively be used to study this nonlinear Muskingum model. (ii) In the parameter estimation of the Muskingum model, YWLDL Algorithm shows good approximation. (iii) The final points (x_1, x_2, and x_3) are competitive with the final points of similar algorithms.

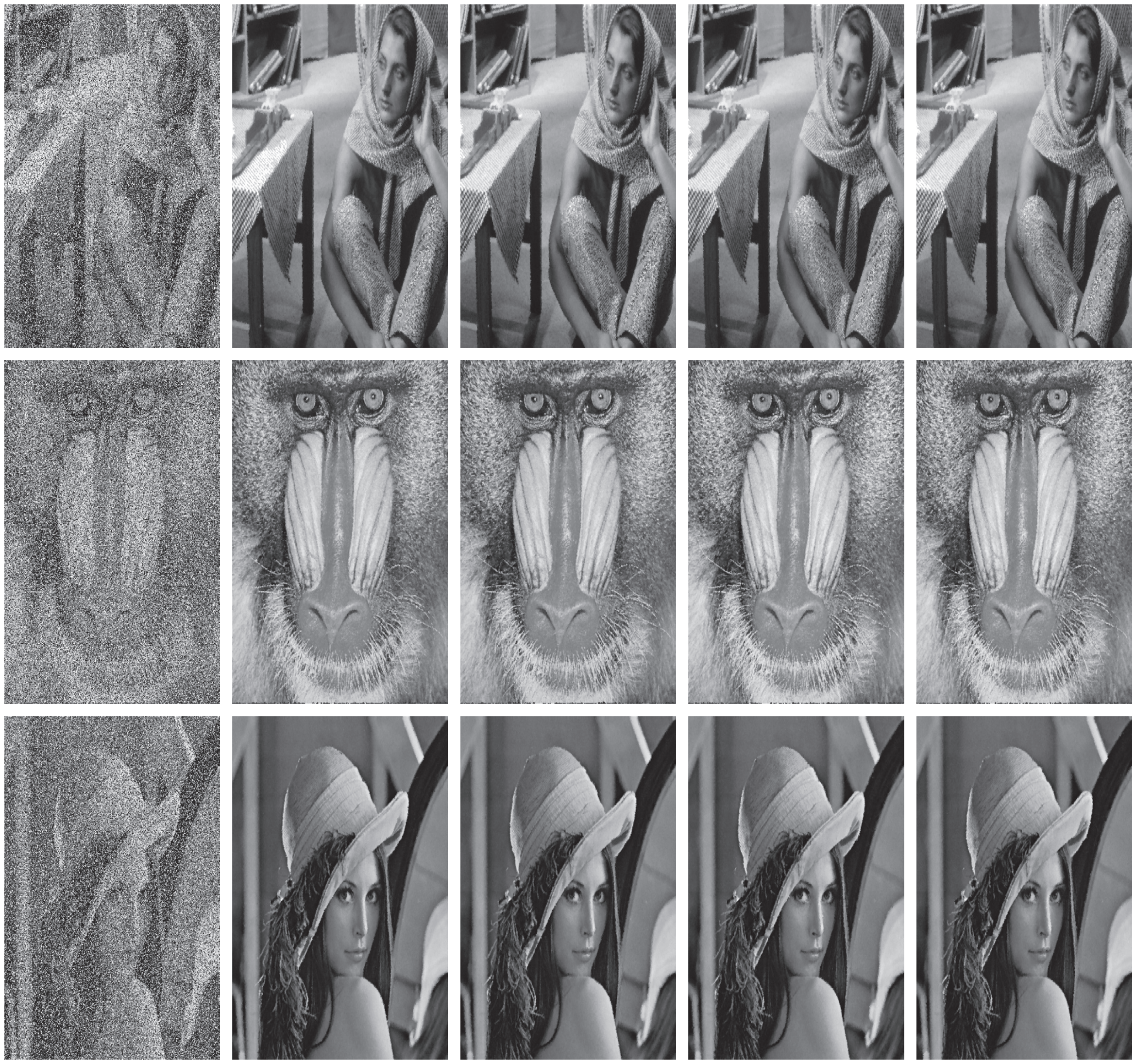

Figure 8: Restoration of the Babara, Baboon, and Lena images by using the HZ algorithm, BG algorithm, DL+ algorithm, and YWLDL algorithm. From left to right: a noisy image with 45% salt-and-pepper noise, restorations obtained by minimizing z with the HZ algorithm, BG algorithm, and DL+ algorithm and YWLDL algorithm.

4.3. Image Restoration Problems. In this section, we will use the proposed algorithm to recover the original image from the image destroyed by impulse noise. It has important practical significance in the field of optimization. The selection of parameters is similar to that in the section of Numerical Experiments. The stop condition is $(| \|f_{k+1}\| - \|f_k\| | / \|f_k\|) < 10^{-3}$ or $(\|x_{k+1} - x_k\| / \|x_k\|) < 10^{-3}$ holds. The experiments choose Barbara (512×512), Baboon (512×512), and Lena (512×512) as the test images. The detailed performance results are given in Figures 7–9. It is not difficult to observe that both HZ algorithm, BG algorithm, and DL+ algorithm and YWLDL algorithm are tested in the image restoration of tested images. The expenditure of the CPU time is listed in Table 3 to compare the YWLDL algorithm with the other algorithms.

From Table 3 and Figures 7–9, we may draw the following conclusions: (i) It pretty obviously takes less time to restore an image with 25% noise than one with 65% noise. (ii) When the salt-and-pepper noise increases, the cost to restore the image increases. (iii) The YWLDL algorithm shows advantage to the HZ algorithm, BG algorithm, and DL+ algorithm in image restoration problems.

5. Conclusion

In this paper, the YWLDL conjugate gradient algorithm that combines with the modified WWP line search technique is proposed. The theory of other methods for constructing parameter t is one of the interesting works, and we think that some modified DL methods with other parameter t under

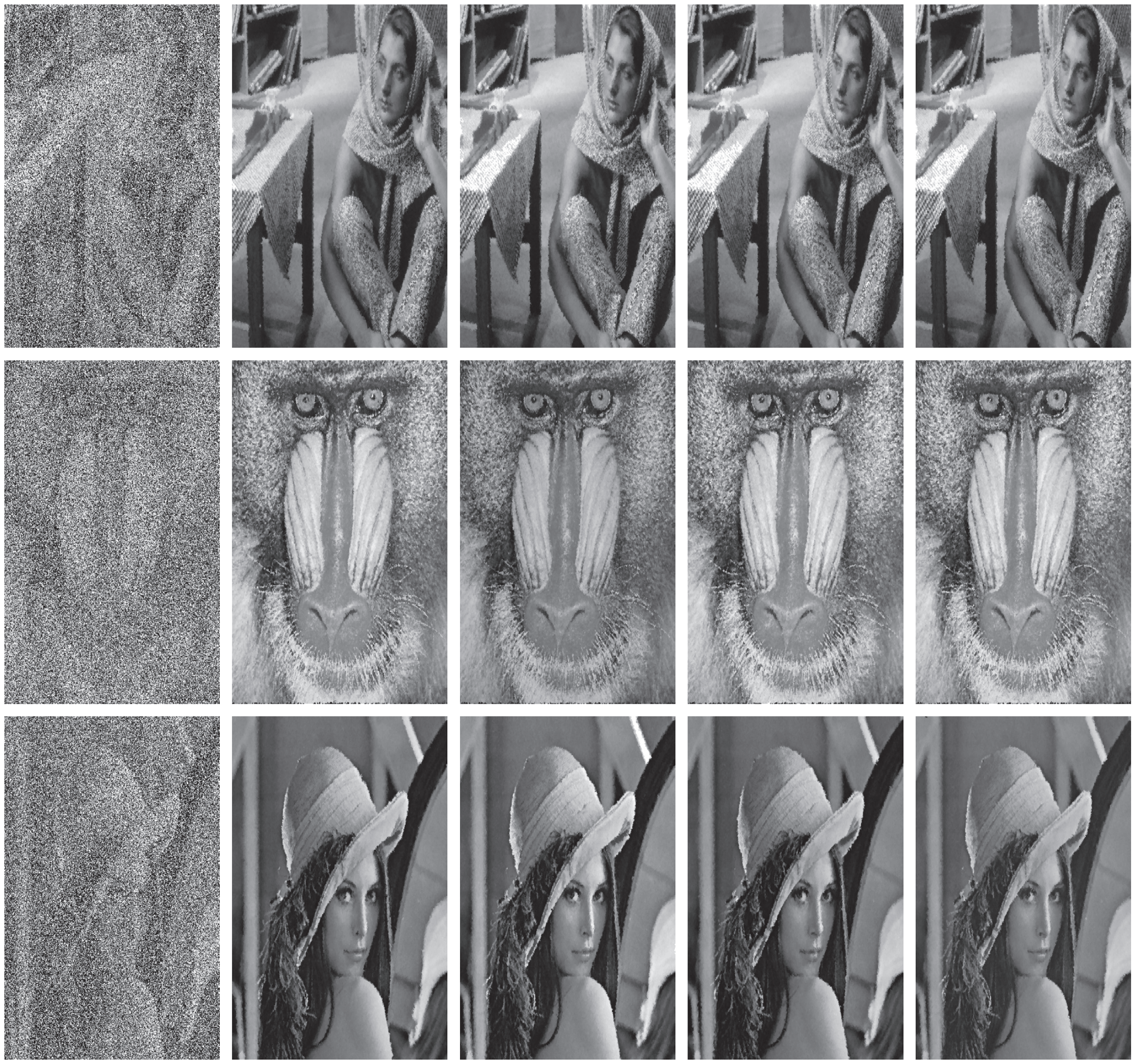

Figure 9: Restoration of the Babara, Baboon, and Lena images by the HZ algorithm, BG algorithm, DL+ algorithm, and YWLDL algorithm. From left to right: a noisy image with 65% salt-and-pepper noise, restorations obtained by minimizing z with the HZ algorithm, BG algorithm, and DL+ algorithm and YWLDL algorithm.

Table 3: The CPU time of HZ algorithm, BG algorithm, DL+ algorithm, and YWLDL algorithm in seconds.

25% noise	Barbara	Baboon	Lena	Total
HZ algorithm	0.84375	1.218750	1.046875	3.109375
BG algorithm	1.000000	0.921875	1.031250	2.953125
DL+ algorithm	1.015625	0.953125	0.953125	2.921875
YWLDL algorithm	0.937500	0.906250	0.937500	2.78125
45% noise	Barbara	Baboon	Lena	Total
HZ algorithm	1.328125	1.312500	1.343750	3.984375
BG algorithm	1.359375	1.285250	1.187500	3.832125
DL+ algorithm	1.250000	1.265625	1.28125	3.796875
YWLDL algorithm	1.203125	1.125000	1.15625	3.484375
65% noise	Barbara	Baboon	Lena	Total
HZ algorithm	1.812500	1.906250	1.671875	5.390625
BG algorithm	1.921875	1.546875	1.78150	5.25025
DL+ algorithm	1.65625	1.578125	1.765625	5.00000
YWLDL algorithm	1.546875	1.500000	1.562500	4.609375

the modified WWP line search technique can also solve the Muskingum model, and we may do this work in the future.

Data Availability

The data used to support the findings of this study are included within the article.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (Grant No. 11661009), the High Level Innovation Teams and Excellent Scholars Program in Guangxi Institutions of Higher Education (Grant No. [2019] 52), the Guangxi Natural Science Foundation (2020GXNSFAA159069), and the Guangxi Natural Science Key Fund (No. 2017GXNSFDA198046).

References

[1] E. G. Birgin and M. Martínez, "A spectral conjugate gradient method for unconstrained optimization," *Applied Mathematics and Optimization*, vol. 43, no. 2, pp. 117–128, 2001.
[2] A. I. Cohen, "Rate of convergence of several conjugate gradient algorithms," *SIAM Journal on Numerical Analysis*, vol. 9, no. 2, pp. 248–259, 1972.
[3] Z. Dai and H. Zhu, "A modified Hestenes-Stiefel-type derivative-free method for large-scale nonlinear monotone equations," *Mathematics*, vol. 168, pp. 135–164, 2020.
[4] G. Yuan, Z. Wei, and X. Lu, "Global convergence of BFGS and PRP methods under a modified weak Wolfe-Powell line search," *Applied Mathematical Modelling*, vol. 47, pp. 811–825, 2017.
[5] Y.-X. Yuan, "Analysis on the conjugate gradient method," *Optimization Methods and Software*, vol. 2, no. 1, pp. 19–29, 1993.
[6] Y.-H. Dai, "New properties of a nonlinear conjugate gradient method," *Numerische Mathematik*, vol. 89, no. 1, pp. 83–98, 2001.
[7] J. Nocedal and J. Wright, "Numberical optimization," *Springer Series in Operations Research*, Springer, Berlin, Germany, 2nd edition, 2006.
[8] Z. Shi, "Restricted PR conjugate gradient method and its global convergence," *Advances in Mathematics*, vol. 31, no. 1, pp. 47–55, 2002.
[9] G. Yuan, T. Li, and W. Hu, "A conjugate gradient algorithm for large-scale nonlinear equations and image restoration problems," *Applied Numerical Mathematics*, vol. 147, pp. 129–141, 2020.
[10] G. Yuan, J. Lu, and Z. Wang, "The PRP conjugate gradient algorithm with a modified WWP line search and its application in the image restoration problems," *Applied Numerical Mathematics*, vol. 152, pp. 1–11, 2020.
[11] G. Yuan, X. Wang, and Z. Sheng, "Family weak conjugate gradient algorithms and their convergence analysis for nonconvex functions," *Numerical Algorithms*, vol. 84, no. 3, pp. 935–956, 2020.
[12] E. Polak and G. Ribière, "Note sur la convergence de méthodes de directions conjuguées," *Revue Française d'Informatique et de Recherche Opérationnelle. Série Rouge*, vol. 3, no. 16, pp. 35–43, 1969.
[13] B. T. Polyak, "The conjugate gradient method in extremal problems," *USSR Computational Mathematics and Mathematical Physics*, vol. 9, no. 4, pp. 94–112, 1969.
[14] R. Fletcher and C. M. Reeves, "Function minimization by conjugate gradients," *The Computer Journal*, vol. 7, no. 2, pp. 149–154, 1964.
[15] M. R. Hestenes and E. Stiefel, "Methods of conjugate gradients for solving linear systems," *Journal of Research of the National Bureau of Standards*, vol. 49, no. 6, pp. 409–436, 1952.
[16] R. Fletcher, *Practical Method of Optimization, Vol I: Unconstrained Optimization*, Wiley, New York, NY, USA, 2nd edition, 1997.
[17] Y. Liu and C. Storey, "Efficient generalized conjugate gradient algorithms part 1: theory," *Journal of Optimization Theory and Applications Mathematics*, vol. 10, pp. 177–182, 2000.
[18] Y. H. Dai and Y. Yuan, "A nonlinear conjugate gradient method with a strong global convergence property," *SIAM Journal on Optimization*, vol. 10, no. 1, pp. 177–182, 1999.
[19] M. J. D. Powell, "Convergence properties of algorithms for nonlinear optimization," *SIAM Review*, vol. 28, no. 4, pp. 487–500, 1986.
[20] Y.-H. Dai and Z. Liao, "New conjugacy conditions and related nonlinear conjugate gradient methods," *Applied Mathematics and Optimization*, vol. 43, no. 1, pp. 87–101, 2001.
[21] W. W. Hager and H. Zhang, "A new conjugate gradient method with guaranteed descent and an efficient line search," *SIAM Journal on Optimization*, vol. 16, no. 1, pp. 170–192, 2005.
[22] S. Babaie-Kafaki and R. Ghanbari, "A class of adaptive Dai-Liao conjugate gradient methods based on the scaled memoryless BFGS update," *4OR*, vol. 15, no. 1, pp. 85–92, 2017.
[23] Y.-H. Dai and C.-X. Kou, "A nonlinear conjugate gradient algorithm with an optimal property and an improved wolfe line search," *SIAM Journal on Optimization*, vol. 23, no. 1, pp. 296–320, 2013.
[24] M. Fatemi, "A new efficient conjugate gradient method for unconstrained optimization," *Journal of Computational and Applied Mathematics*, vol. 300, pp. 207–216, 2016.
[25] M. Fatemi, "An optimal parameter for Dai-Liao family of conjugate gradient methods," *Journal of Optimization Theory and Applications*, vol. 169, no. 2, pp. 587–605, 2016.
[26] Z. Wei, G. Yu, G. Yuan, and Z. Lian, "The superlinear convergence of a modified BFGS-type method for unconstrained optimization," *Computational Optimization and Applications*, vol. 29, no. 3, pp. 315–332, 2004.
[27] Z. Wei, G. Li, and L. Qi, "New quasi-Newton methods for unconstrained optimization problems," *Applied Mathematics and Computation*, vol. 175, no. 2, pp. 1156–1188, 2006.
[28] Z. Aminifard and S. Babaie-Kafaki, "An optimal parameter choice for the Dai-Liao family of conjugate gradient methods by avoiding a direction of the maximum magnification by the search direction matrix," *A Quarterly Journal of Operations Research*, vol. 17, pp. 1–14, 2019.
[29] S. Babaie-Kafaki, "On the sufficient descent condition of the Hager-Zhang conjugate gradient methods," *4OR*, vol. 12, no. 3, pp. 285–292, 2014.
[30] G. Yuan, Z. Wei, and Y. Yang, "The global convergence of the Polak-Ribière-Polyak conjugate gradient algorithm under inexact line search for nonconvex functions," *Journal of*

Computational and Applied Mathematics, vol. 362, pp. 262–275, 2019.

[31] Y. Yuan and W. Sun, *Theory and Methods of Optimization*, Science Press of China, Beijing, China, 1999.

[32] E. D. Dolan and J. J. Moré, "Benchmarking optimization software with performance profiles," *Mathematical Programming*, vol. 91, no. 2, pp. 201–213, 2002.

[33] A. Ouyang, L. Liu, Z. Sheng, and F. Wu, "A class of parameter estimation methods for nonlinear Muskingum model using hybrid invasive weed optimization algorithm," *Mathematical Problems Engingering*, vol. 2015, Article ID 573894, 15 pages, 2015.

[34] A. Ouyang, Z. Tang, K. Li, A. Sallam, and E. Sha, "Estimating parameters of Muskingum model using an adaptive hybrid PSO algorithm," *International Journal of Pattern Recognition and Artificial Intellgence*, vol. 28, p. 29, Article ID 1459003, 2014.

[35] Z. W. Geem, "Parameter estimation for the nonlinear Muskingum model using the BFGS technique," *Journal of Irrigation and Drainage Engineering*, vol. 132, no. 5, pp. 474–478, 2006.