Journal of Applied Mathematics and Computing (2024) 70:3257–3280
https://doi.org/10.1007/s12190-024-02088-2

ORIGINAL RESEARCH

A conjugate gradient algorithm without Lipchitz continuity and its applications

Huiyun Liu[1] · **Haishan Feng**[1]

Received: 12 March 2024 / Revised: 2 April 2024 / Accepted: 8 April 2024 /
Published online: 3 May 2024

Abstract
An improved conjugate gradient algorithm is proposed that does not rely on the line search rule and automatically achieves sufficient descent and trust region qualities. It is applicable to solve unconstrained problems and large-scale nonsmooth problems. Furthermore, it demonstrates global convergence properties without the need for Lipschitz continuity conditions. Numerical experiments on nonconvex unconstrained problems and large scale nonsmooth convex optimization problems demonstrate the effectiveness and efficiency of the proposed algorithm compared with the same structural algorithm. Finally, the new algorithm is applied to Muskingum model solving in engineering problems and image restoration, which shows the prospect of the new algorithm.

Keywords Conjugate gradient · Global convergence · Nonsmooth · Image restoration

Mathematics Subject Classification 90-10

1 Introduction

Consider optimization problem:

$$\min\{q(x)|x \in \mathbb{R}^n\}, \tag{1}$$

✉ Haishan Feng
516791535@qq.com

Huiyun Liu
2206301021@st.gxu.edu.cn

[1] School of Mathematics and Information Science, Center for Applied Mathematics of Guangxi (Guangxi University), Guangxi University, Nanning 530004, Guangxi, China

where $q(x) : \mathbb{R}^n \to \mathbb{R}$, is a convex function that possibly is nonsmooth. With the rapid advancement of information technology, optimization techniques are being applied to various aspects of production and daily life. The need to investigate large-scale nonsmooth convex optimization problems has become increasingly urgent. This paper presents an algorithm for addressing large-scale nonsmooth problems, using the problem of image restoration in image processing as an illustrative example. Image restoration refers to observed images are reconstructed into their original, yet unknown, real forms by establishing image degradation models. In the practical processes of image generation, storage, and transmission, image quality deterioration is an issue that cannot be avoided due to technological constraints and objective factors. To obtain more valuable information and ensure that images meet the high-quality research and application standards, image restoration has become a necessary technique. The following are common image degradation models

$$\beta = \Lambda x + \epsilon,$$

where $\beta \in \mathbb{R}^m$ and $x \in \mathbb{R}^n$ respectively correspond to the observed and original images, Λ is an $m \times n$ matrix responsible for blurring, $\epsilon \in \mathbb{R}^m$ is the noise term. To solve ϵ, We first solve problem

$$\min_{x \in \mathbb{R}^n} \|\Lambda x + \beta\|^2 + \lambda \|Dx\|_1, \tag{2}$$

where λ as regularization parameter, D as linear operator. In this paper, $\|\cdot\|$ denotes Euclidean norm, $\|\cdot\|_1$ is l_1 norm. Since l_1 norm is nonsmooth, (2) described above falls into the category of nonsmooth convex optimization problems, which are typically of large scale.

Compared to unconstrained problems, the objective function of non-smooth problems may include discontinuous components, which pose challenges to problem-solving. Furthermore, in the current conjugate algorithms, the assumption of Lipschitz continuity is required to ensure bounded gradient variations. This condition is a key requirement in the convergence analysis of line search. However, in practice, there are instances where the Lipschitz continuity hypothesis may not hold or cannot be readily verified due to complexity of the objective function or limitations in obtaining accurate gradient information. Some algorithms [1, 2] that do not require Lipschitz continuity have been proposed. To overcome these constraints, we present a novel algorithm in this paper. It tackles unconstrained and nonsmooth issues employing the Moreau-Yosida regularization technique.The new algorithm possesses three characteristics:

- It fulfills the prerequisites of a sufficient descent and trust region properties, eliminates need to specify a step size.
- The algorithm's performance is improved by incorporating function value and gradient information into the search direction.
- The algorithm integrates Wake-Wolfe-Powell (WWP) linear search(5)(6) and exhibits global convergence for nonLipschitz continuous nonconvex problems and nonsmooth functions.

In Sect. 2, an algorithm is presented which has some good properties for solving unconstrained optimization problems. In Sect. 3, the algorithm is used to solve nonsmooth problems. Section 4 states numerical findings of algorithms for unconstrained and nonsmooth problems. Additionally, we show the algorithm's numerical performance in image restoration problems. The paper ends with a summary in Sect. 5, outlining possible directions for further investigation.

2 Unconstrained optimization

Regarding problem (1), we first discuss the situation under smooth conditions, i.e.

$$\min\{f(x)|x \in \mathbb{R}^n\}, \tag{3}$$

with $f(x) : \mathbb{R}^n \to \mathbb{R}$ is a continuously differentiable nonconvex function. For solving (3), conjugate gradient(CG) algorithm proves to be highly effective, particularly when its dimension n is of a large scale. CG algorithm generates an iterative sequence by the following formula:

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

where $\alpha_k > 0$ is a steplength determined by line search, the search direction d_k is generated by

$$d_k = -h_k + \beta_k d_{k-1}, \quad d_0 = -g_0,$$

where h_k is the gradient of $f(x_k)$. CG algorithm can be classified according to the choice of parameter β_k, among them, Polak-Ribière-Polyak(PRP) methods has advantages in dealing with large-scale problems because of its high computational efficiency and small storage capacity. Despite employing strong Wolfe line search, achieving global convergence for PRP methods with general nonlinear functions remains challenging. And its primary applications are currently mostly limited to dealing with smooth problems. The BFGS algorithm constructs an approximate Hessian matrix to expedite the convergence of the objective function, the search direction d_k is generated by

$$d_k = -B_k^{-1} h_k,$$

where B_k is the approximate Hessian matrix. BFGS algorithm has made some progress on global convergence of general functions under WWP line search [3]. But its applicability is challenged in large-scale computations due to storage constraints. An adaptive memoryless BFGS method [4, 5] is proposed that adaptively adjusts Hessian matrix approximation, avoiding the need to store Hessian matrix or its inverse. In an attempt to advance the CG and BFGS techniques, a family of conjugate gradient algorithm is

introduced, which d_k can closely aligned with scaled memoryless BFGS method [6]

$$d_k^{\mathrm{DK}}(\gamma_k) = -h_k + \left[\frac{h_k^T y_{k-1}}{d_{k-1}^T y_{k-1}} - \left(\gamma_k + \frac{\|y_{k-1}\|^2}{s_{k-1}^T y_{k-1}} - \frac{s_{k-1}^T y_{k-1}}{\|s_{k-1}\|^2}\right)\frac{h_k^T s_{k-1}}{d_{k-1}^T y_{k-1}}\right] d_{k-1},$$

where $s_{k-1} = x_k - x_{k-1}$, h_k represents gradient of $f(x)$, γ_k denotes the self-scaling parameter and the optimal choice for efficiency is

$$\gamma_k = \frac{s_{k-1}^T y_{k-1}}{\|s_{k-1}\|^2}.$$

Li [7, 8] made some modifications based on the Dai-Kou algorithm, resulting in different numerical outcomes. To enhance convergence, we implement a novel technique ensuring that the search direction d_k adheres to trust region property automatically. This plays a crucial role for establishing global convergence.

Besides directly modifying iteration formula for d_k it is also a viable approach to make adjustments to the quasi-Newton equation. Yuan [9] designs a new quasi-Newtonian equation so that y_k^m has information about the gradient of the function, and also about the function itself

$$B_k s_{k-1} = y_{k-1}^m,$$

where $y_{k-1}^m = y_{k-1} + \frac{\max\{\varrho_{k-1},0\}}{\|s_{k-1}\|^2} s_{k-1}$, $\varrho_{k-1} = (h_k + h_{k-1})^T s_{k-1} + 2(f_{k-1} - f_k)$, denote $f_k = f(x_k)$. Compared with the previous y_k, y_k^m enables the new algorithm to obtain satisfactory results in less iterations and shorter operation time, and has better numerical performance and theoretical results.

Drawing inspiration from the previously mentioned techniques, a novel hybrid conjugate gradient algorithm is introduced. d_k is as follows

$$d_k = -h_k + \left(\frac{h_k^T y_{k-1}^m}{\omega_k} - \frac{\|y_{k-1}^m\|^2 r_k}{\omega_k^2}\right) d_{k-1} + \frac{t_k r_k}{\omega_k} y_{k-1}^m, \quad d_0 = -h_0, \tag{4}$$

where $\omega_k = c_1\|d_{k-1}\|\|y_{k-1}^m\| + c_2\|h_k\|^2$, $r_k = h_k^T d_{k-1}$. By choosing the first term of ω_k, d_k is endowed with a trust region property, and the second term sets a lower bound for ω_k, guaranteeing global convergence. Interestingly, ω_k is never equal to zero, which guarantees that d_k is well-defined.

The algorithm is outlined below

Algorithm 1 BPRP(unconstrained optimization)

Step 0. Input and Initialization. Select $k = 0$ and $x_0 \in \mathbb{R}^n$, $d_0 = -h_0$, $\epsilon > 0$, $0 < \mu < \nu < 1$, $c_1 > 0$, $c_2 > 0$, and $0 \leq t_k \leq \tau < 1$.
Step 1. Calculate $\|h_k\|$, if $\|h_k\| \leq \epsilon$, Terminate the iteration; otherwise turn to Step 2 to continue the iterative calculation.
Step 2. Select α_k that satisfies the WWP line search

$$f(x_k + \alpha_k d_k) \leq f_k + \mu \alpha_k h_k^T d_k, \tag{5}$$

$$h(x_k + \alpha_k d_k)^T d_k \geq \nu h_k^T d_k, \tag{6}$$

with $0 < \mu < \frac{1}{2}$, $\mu < \nu < 1$ are constants.
Step 3. Update iteration point x_{k+1} can be obtained using $x_{k+1} = x_k + \alpha_k d_k$.
Step 4. Calculate d_k with (4).
Step 5. Turn to step 1 for the following iteration, let $k = k + 1$.

Assumption 1 Consider $f(x)$ continuously differentiable, and

$$\Psi = \{x | f(x) \leq f_0\}$$

is bounded.

Lemma 1 *All d_k follows sufficient descent property*

$$h_k^T d_k \leq -z\|h_k\|^2, \tag{7}$$

trust region property

$$\|d_k\| \leq c\|h_k\|, \tag{8}$$

with $z = 1 - \frac{(1+\tau)^2}{4}$ *and* $c = 1 + \frac{1+\tau}{c_1} + \frac{1}{c_1^2}$.

Proof Given that $d_0 = -h_0$, we observe that $h_0^T d_0 = -\|h_0\|^2$, fulfilling (7) for $k = 0$. In light of d_k's definition, we can confirm that for all $k > 0$,

$$\begin{aligned} h_k^T d_k &\leq -\|h_k\|^2 + \frac{(1+t_k)^2}{4}\|h_k\|^2 + \frac{\|y_{k-1}^m\|^2 r_k^2}{\omega_k^2} - \frac{\|y_{k-1}^m\|^2 r_k^2}{\omega_k^2} \\ &\leq -\left(1 - \frac{(1+\tau)^2}{4}\right)\|h_k\|^2, \end{aligned}$$

which implies that (7) holds.

When $k = 0$, using $d_0 = -h_0$ as a starting point, we obtain $\|d_0\| = \|h_0\|$, implying (8). For $k > 0$, we can use (7) and Cauchy-Schwarz inequality to establish (8). Assume $y_{k-1}^m \neq 0$, consider

$$\omega_k \geq c_1 \|d_{k-1}\| \|y_{k-1}^m\|.$$

Then, we have

$$\begin{aligned}\left|\frac{h_k^T y_{k-1}^m}{\omega_k}-\frac{\|y_{k-1}^m\|^2 r_k}{\omega_k^2}\right| &\le \frac{\|h_k\|\|y_{k-1}^m\|}{\omega_k}+\frac{\|y_{k-1}^m\|^2\|h_k\|\|d_{k-1}\|}{\omega_k^2}\\ \le \frac{\|h_k\|\|y_{k-1}^m\|}{c_1\|d_{k-1}\|\|y_{k-1}^m\|}+\frac{\|y_{k-1}^m\|^2\|h_k\|\|d_{k-1}\|}{(c_1\|d_{k-1}\|\|y_{k-1}^m\|)^2} &= \left(\frac{1}{c_1}+\frac{1}{c_1^2}\right)\frac{\|h_k\|}{\|d_{k-1}\|},\end{aligned} \tag{9}$$

and

$$\left|t_k\frac{r_k}{\omega_k}\right| \le \tau\frac{\|h_k\|\|d_{k-1}\|}{c_1\|d_{k-1}\|\|y_{k-1}^m\|}=\tau\frac{\|h_k\|}{c_1\|y_{k-1}^m\|}. \tag{10}$$

Combining (9) (10) (4), we yield

$$\begin{aligned}\|d_k\| &\le \|h_k\|+\left(\frac{1}{c_1}+\frac{1}{c_1^2}\right)\frac{\|h_k\|}{\|d_{k-1}\|}\|d_{k-1}\|+\tau\frac{\|h_k\|}{c_1\|y_{k-1}^m\|}\|y_{k-1}^m\|\\ &\le \left(1+\frac{1+\tau}{c_1}+\frac{1}{c_1^2}\right)\|h_k\|.\end{aligned}$$

The proof is completed. □

Theorem 1 *Combining Algorithm 2 and Assumption 1, then*

$$\liminf_{k\to\infty}\|h_k\|=0. \tag{11}$$

Proof Proofs by contradiction. If (11) is incorrect, suggests that exist constant $\epsilon > 0$ and have

$$\|h_k\| \ge \epsilon. \tag{12}$$

Sequence $\{\alpha_k\}$ is assumed to converge and denote $\overline{\alpha}=\limsup_{k\to\infty}\alpha_k$. We observe that $\overline{\alpha}\ge 0$. Accordingly, the discussion follows.

Case (i): $\overline{\alpha} > 0$. When $k_i > k$, exists subsequence $\{\alpha_{k_i}\}$ and constant $\xi > 0$ satisfies

$$\lim_{i\to\infty}\alpha_{k_i} > \xi,$$

By applying (5) and summing up from $k = 0$ to ∞ yields

$$-\mu\sum_{k=1}^{\infty}\alpha_k h_k^T d_k < f_0-\lim_{k\to\infty} f_k < \infty,$$

then

$$\lim_{i\to\infty} -\alpha_{k_i} g_{k_i}^T d_{k_i} = 0.$$

By (7) and (12), thus

$$\lim_{i\to\infty}(1-\frac{(1+\tau)^2}{4})\alpha_{k_i}\|\epsilon\|^2 \le \lim_{i\to\infty}(1-\frac{(1+\tau)^2}{4})\alpha_{k_i}\|h_{k_i}\|^2 \le \lim_{i\to\infty} -\alpha_{k_i} h_{k_i}^T d_{k_i} = 0.$$

This implies

$$\lim_{i\to\infty} \alpha_{k_i} = 0,$$

which contradicts our assumption. Therefore, the (11) is true.

Case (ii) $\overline{\alpha} = 0$. From (6)

$$h(x_{k_l} + \alpha_{k_l} d_{k_l})^T d_{k_l} - \nu h_{k_l}^T d_{k_l} \ge 0.$$

So we obtain

$$\lim_{l\to\infty} h(x_{k_l} + \alpha_{k_l} d_{k_l})^T d_{k_l} - \nu h(x_{k_l})^T d_{k_l} = (1-\nu)h(x^*)^T d(x^*) \ge 0. \tag{13}$$

Combine with (7) and (12), then

$$h(x^*)^T d(x^*) \le -z\|h(x^*)\|^2 \le -z\epsilon < 0. \tag{14}$$

This is contradictory to (32). □

According to Theorem 1, we assume $\lim_{k\to\infty} x_k = \hat{x}$. Then, under the following additional assumptions, we further discuss the convergence rate of BPRP algorithm.

Assumption 2 Assuming f that is uniformly convex and has continuous second derivatives in $\mathbb{R}^n$, gradient h satisfies Lipschitz continuity. So f has unique minimal point $\hat{x}$ with minimum value $\hat{f}$, for all $x \in \mathbb{R}^n$ satisfying

$$\frac{1}{2}\varphi \left\|x-\hat{x}\right\|^2 \le f(x) - \hat{f} \le \frac{1}{2}\phi \left\|x-\hat{x}\right\|^2, \tag{15}$$

and

$$\varphi \left\|x_k-\hat{x}\right\|^2 \le \|h_k\|^2 \le \phi \left\|x_k-\hat{x}\right\|^2, \tag{16}$$

where $0 < \varphi < \phi$ are constants.

Lemma 2 *$\{x_k\}$ is obtained by BPRP algorithm. If the conditions in Assumption 1 and (15) (16) are satisfied, for all $k \geq 0$ one can derive*

$$\alpha_k \geq \vartheta, \tag{17}$$

where $\vartheta > 0$ a constant.

Proof Denote

$$\Omega_{k-1} = \int_0^1 \nabla^2 f(x_{k-1} + \zeta s_{k-1}) d\zeta,$$

By to mean-value theorem,

$$h_k - h_{k-1} = \Omega_{k-1} s_{k-1} = \Omega_{k-1} \alpha_{k-1} d_{k-1}.$$

From (6), we have

$$h\left(x_k + \alpha_k d_k\right)^T d_k = \left(h_k + \alpha_k d_k \int_0^1 \nabla^2 f\left(x_k + \zeta \alpha_k d_k\right) d\zeta\right)^T d_k \geq v h_k^T d_k,$$

which implies

$$\alpha_k d_k^T \int_0^1 \nabla^2 f\left(x_k + \zeta \alpha_k d_k\right) d\zeta d_k \geq (v-1) h_k^T d_k. \tag{18}$$

Based on Assumption 2, then $\varphi \|d\|^2 \leq d^T \nabla^2 f(x) d \leq \phi \|d\|^2$, combining with (18) have $\phi \alpha_k \|d_k\|^2 \geq (v-1) h_k^T d_k$. Based on (8), we have

$$\begin{aligned} \alpha_k &\geq \frac{(v-1) h_k^T d_k}{\phi \|d_k\|^2} \geq \frac{(1-v)}{\phi} \frac{\left(1 - \frac{(1+\tau)^2}{4}\right) \|h_k\|^2}{\|d_k\|^2} \\ &\geq \frac{(1-v)}{\phi} \left(1 - \frac{(1+\tau)^2}{4}\right) \left(1 + \frac{1+\tau}{c_1} + \frac{1}{c_1^2}\right)^{-2} \triangleq \overline{\vartheta}. \end{aligned}$$

(17) is obtained by setting $\vartheta = min\{1, \overline{\vartheta}\}$. □

Theorem 2 *According to Assumption 2, $\{x_k\}$ converges to $\hat{x}$, satisfies*

$$\|x_k - \hat{x}\| \leq \hat{b} \sigma^k, \tag{19}$$

where $\hat{b} > 0$ and $0 < \sigma < 1$ are constants.

Proof From (5) in WWP, (7)(15) and (16)

$$\begin{aligned} f_{k+1} - \hat{f} &\le f_k + \mu\alpha_k h_k^T d_k - \hat{f} \\ &\le f_k - (1 - \frac{(1+\tau)^2}{4})\mu\psi\frac{2\varphi}{\phi}\left(f_k - \hat{f}\right) - \hat{f} \\ &= \left(1 - \left(1 - \frac{(1+\tau)^2}{4}\right)\mu\psi\frac{2\varphi}{\phi}\right)\left(f_k - \hat{f}\right). \end{aligned}$$

Setting $\sigma = \left(1 - (1 - \frac{(1+\tau)^2}{4})\mu\psi\frac{2\varphi}{\phi}\right)^{\frac{1}{2}}$, so have

$$f_k - \hat{f} \le \sigma^2\left(f_{k-1} - \hat{f}\right) \le \cdots \le \sigma^{2k}\left(f_0 - \hat{f}\right).$$

Combining (15), then

$$\left\|x_k - \hat{x}\right\|^2 \le \frac{2}{\varphi}\left(f_k - \hat{f}\right) \le \frac{2}{\varphi}\left(f_0 - \hat{f}\right)\sigma^{2k},$$

which shows that (19) holds, where $\hat{b} = \left(\frac{2}{\varphi}\left(f_0 - \hat{f}\right)\right)^{\frac{1}{2}}$. □

3 Nonsmooth problem

Adding regularization term to nonsmooth convex problem (1)

$$\min_{x\in\mathbb{R}^n} F(x) \triangleq \min_{r\in\mathbb{R}^n}\{q(r) + \frac{1}{2\chi}\|r - x\|^2\}, \tag{20}$$

where $F : \mathbb{R}^n \to \mathbb{R}$, $\chi > 0$. (20) is considered equivalent to (1). Setting $\Theta(r, x) = q(r) + \frac{1}{2\chi}\|r - x\|^2$, and $\ell(x) = \text{argmin}_r\, \Theta(r, x)$. For every x, $\Theta(\cdot, x)$ exhibits strong convexity. So F is denoted as

$$F(x) = q(\ell(x)) + \frac{1}{2\chi}\|\ell(x) - x\|^2.$$

F has the following properties:

(i) It is Finite-valued and convex. Denote gradient of F as

$$\hat{h}(x) = \nabla F(x) = \frac{x - \ell(x)}{\chi},$$

it exists in $\mathbb{R}^n$ and continuous

(ii) When $\hat{h}(x) = 0$, i.e. $\ell(x) = x$, (1) has unique solution. By minimizing $\Theta(r)$, can get $\ell(x)$, which can expresses $F(x)$ and $\hat{h}(x)$. But finding exact minimizer $\Theta(r)$ can be quite challenging or even unfeasible, $F(x)$ and $\hat{h}(x)$ are difficult to express explicitly. We can use the finite-valued of $F(x)$. For any $\delta > 0$, exists $\ell^a(x, \delta)$ that satisfies

$$q\left(\ell^a(x,\delta)\right) + \frac{1}{2\chi}\left\|\ell^a(x,\delta) - x\right\|^2 \le F(x) + \delta. \tag{21}$$

Therefore, the estimates of $F(x)$ and $\hat{h}(x)$ are expressed as

$$F^a(x,\delta) = q\left(\ell^a(x,\delta)\right) + \frac{1}{2\lambda}\left\|\ell^a(x,\delta) - x\right\|^2, \tag{22}$$

$$\hat{h}(x)^a(x,\delta) = \frac{x - \ell^a(x,\delta)}{\chi}. \tag{23}$$

For non-differentiable convex function, some useful algorithms are available to obtain $\ell^a(x, \delta)$, as introduced in [10].

Proposition 3 *Assuming $F^a(x, \delta)$ and $\hat{h}(x)^a(x, \delta)$ are obtained from (22)(23), where $\ell^a(x, \delta)$ satisfies (21), we can deduce [11]*

$$\begin{aligned} &F(x) \le F^a(x,\delta) \le F(x) + \delta, \\ &\left\|\ell^a(x,\delta) - \ell(x)\right\| \le \sqrt{2\chi\delta}, \\ &\left\|\hat{h}^a(x,\delta) - \hat{h}(x)\right\| \le \sqrt{2\delta/\chi}. \end{aligned} \tag{24}$$

This means that $F^a(x, \delta)$ and $\hat{h}^a(x, \delta)$ can be considered to be close enough approximations of $F(x)$ and $\hat{h}(x)$.

Combine above conditions and discussion in Sect. 2, iterative formula for d_k is as follows:

$$d_k = -\hat{h}^a(x_k,\delta_k) + \left(\frac{\hat{h}^a(x_k,\delta_k)^T y_{k-1}^m}{\varpi_k} - \frac{\|y_{k-1}^m\|^2 \tilde{h}_k}{\varpi_k^2}\right) d_{k-1} + \frac{t_k \tilde{h}_k}{\varpi_k} y_{k-1}^m, \tag{25}$$

where $\varpi_k = c_1\|d_{k-1}\|\|y_{k-1}^m\| + c_2\|\hat{h}_k\|^2, \tilde{r}_k = \hat{h}^a(x_k,\delta_k)^T d_{k-1}$, with $0 \le t_k \le \tau < 1$. d_k also exhibits sufficient descent and trust region properties, i.e.

$$\begin{aligned} &\hat{h}^a(x_k,\delta_k)^T d_k \le -\tilde{z}\|\hat{h}^a(x_k,\delta_k)\|^2, \\ &\|d_k\| \le \tilde{c}\|\hat{h}^a(x_k,\delta_k)\|, \end{aligned} \tag{26}$$

where $\tilde{z} = 1 - \frac{(1+\tau)^2}{4}$, $\tilde{c} = 1 + \frac{1+\tau}{c_1} + \frac{1}{c_1^2}$. The steps and properties of the algorithm are as follows.

Theorem 4 *$\{x_k\}$ and $\{\hat{h}_k\}$ are generated by Algorithm 2, we have $\lim_{k\to\infty}\|\hat{h}(x_k)\| = 0$, and any accumulation point of x_k is an optimal solution of (1).*

Algorithm 2 BPRP(Nonsmooth problem)

Step 0. Choose x_0, constant $0 < \mu < \nu < 1$, $c_1 > 0$, $c_2 > 0$, $\epsilon > 0$, $0 < \delta_k < 1$, $\lim_{k\to\infty} \delta_k = 0$ and $t_k \in (0, \tau)$, $\tau < 1$. Let $d_0 = -\hat{h}^a(x_0, \delta_0)$, and $k = 0$;
Step 1. When $\|\hat{h}^a(x_k, \delta_k)\| \le \epsilon$, stop algorithm;
Step 2. Choose a suitable value of parameter α_k by WWP line search

$$F^a(x_k + \alpha_k d_k, \delta_{k+1}) \le F^a(x_k, \delta_k) + \mu\alpha_k \hat{h}^a(x_k, \delta_k)^T d_k, \tag{27}$$

$$\hat{h}^a(x_k + \alpha_k d_k, \delta_{k+1})^T d_k \ge \nu \hat{h}^a(x_k, \delta_k)^T d_k. \tag{28}$$

Step 3. Using $x_{k+1} = x_k + \alpha_k d_k$ get next iteration point;
Step 4. d_k is updated according to (25);
Step 5. Set $k = k + 1$, return to Step 1.

Proof **The first part.** Prove that

$$\lim_{k\to\infty} \|\hat{h}^\alpha(x_k, \delta_k)\| = 0. \tag{29}$$

Assuming (29) is incorrect, thus exist subsequence λ, constant $\delta_* > 0$, and $k_* \in \mathbb{Z}$ that satisfy

$$\|\hat{h}^a(x_k, \delta_k)\| \ge \delta_*, \forall \lambda \ni k > k_*. \tag{30}$$

For sequence $\{x_k\}$, x^* is one of its limiting points, then have

$$\lim_{k\in K,\, k\to\infty} x_k = x^*. \tag{31}$$

Consider two cases for discussion.

Case(I). $\limsup_{k\to\infty} \alpha_k > 0$. Thus, there exists subsequence $\{\alpha_{k_j}\}$ such that $\lim_{j\to\infty} \alpha_{k_j} > \tau$, $\tau > 0$ is a constant, $k_j > k$. By (27),

$$F^a(x_k, \delta_k) - F^a(x_k + \alpha_k d_k, \delta_{k+1}) \ge -\mu\alpha_k \hat{h}^a(x_k, \delta_k)^T d_k.$$

Hence

$$-\mu \sum_{k=1}^{\infty} \alpha_k \hat{h}^a(x_k, \delta_k)^T d_k \le F^a(x_0, \delta_0) - \lim_{k\to\infty} F^a(x_k, \delta_k) < \infty.$$

Then

$$\lim_{j\to\infty} -\alpha_{k_j} \hat{h}^a(x_{k_j}, \delta_{k_j})^T d_{k_j} = 0.$$

Combining (26), we can derive

$$\lim_{j\to\infty}(1-\frac{(1+\tau)^2}{4})\alpha_{k_j}\|\epsilon_*\|^2 \le \lim_{j\to\infty} -\alpha_{k_j}\hat{h}^a(x_{k_j},\delta_{k_j})^T d_{k_j} = 0.$$

It means that

$$\lim_{j\to\infty}\alpha_{k_j} = 0,$$

which contradicts the assumption of this case.

Case (II). $\limsup_{k\to\infty}\alpha_k = 0$. From (28) and (26),

$$\hat{h}^a(x_k+\alpha_k d_k,\delta_{k+1})^T d_k - v\hat{h}^a(x_k,\delta_k)^T d_k \ge 0$$

We can conclude:

$$\lim_{j\to\infty}\hat{h}^a(x_{k_j}+\alpha_{k_j}d_{k_j},\delta_{k_j+1})^T d_{k_j} - v\hat{h}^a(x_{k_j},\delta_{k_j})^T d_{k_j} = (1-v)\hat{h}(x^*)^T d(x^*) \ge 0. \tag{32}$$

Combine with (26) and (30), then

$$h(x^*)^T d(x^*) \le -\tilde{z}\|h(x^*)\|^2 \le -\tilde{z}\delta_* < 0. \tag{33}$$

This is contradictory to (32). So (29) holds, and (24) can see that

$$\|\hat{h}^\alpha(x_k,\delta_k) - \hat{h}(x_k)\| \le \sqrt{\frac{2\delta_k}{\chi}}.$$

Combined with $\lim_{k\to\infty}\delta_k = 0$, means

$$\lim_{k\to\infty}\|\hat{h}(x_k)\| = 0. \tag{34}$$

The second part. Verify x_k that converge to a solution of problem (1). By definition of $\hat{h}(x)$, we get $\hat{h}(x_k) = \frac{x_k-\ell(x_k)}{\chi}$. Then, by (34) and (31), $x^* = \ell(x^*)$ holds. Therefore, x^* is an optimal solution of (1). □

4 Numerical experiments

Four experiments are used to evaluate the BPRP algorithm's performance. These results are from a series of experiments, including unconstrained optimization, large-scale nonsmooth problems, Muskingum model in engineering, and Color image

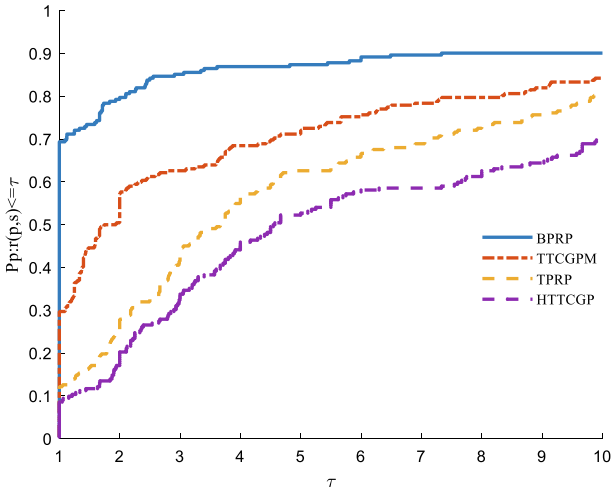

Fig. 1 Performance of algorithms in terms of number of iterations(NI)

restoration. Experimented on Windows 10 computer with 8 GB RAM and 2 GHz CPU.

4.1 Test of unconstrained optimization

Efficacy of BPRP, TTCGPM, TPRP, and HTTCGP algorithms in addressing unconstrained optimization problems becomes evident through the examination of 74 such problem instances. To facilitate a more intuitive comparison regarding the computational efficiency of these algorithms, the analytical tools proposed by Dolan and More [12] are utilized. These tools offer insights into the algorithms' performance metrics.

This investigation focuses specifically on evaluating the computational performance across three dimensions (3000, 6000, and 9000) for the 74 problems. Throughout the experiment, parameters c_1 and c_2 are assigned values of 0.001 and 0.01, respectively. Additionally, the calculation of t_k to the formula $t_k = \min\{\tau, \max\{0, 1 - \frac{y_{k-1}^m{}^T s_{k-1}}{\|y_{k-1}^m\|^2}\}\}$ and $\tau = 0.1$.

The outcomes are presented in Figs. 1, 2, and 3, where τ signifies the reciprocal of the algorithm's performance ratio (NI, NFG, and CPU time) when tackling a specific problem, relative to the optimal performance among all algorithms. The parameter $P_{(p:r(p,s)\leq\tau)}$ on the vertical axis succinctly denotes the proportion of problems successfully resolved out of the total problem set when the algorithm's ratio falls below τ.

By observing the data in Figs. 1, 2, and 3, a trend emerges, the efficacy of the BPRP algorithm in addressing a majority of the examined test problems. Figure 1 shows that the BPRP algorithm has the least number of iterations in 70% problem solving, and can solve 90% of the problems. It can be seen from Fig. 2 that BPRP algorithm calculates function value and gradient value least in 60% of the problems,

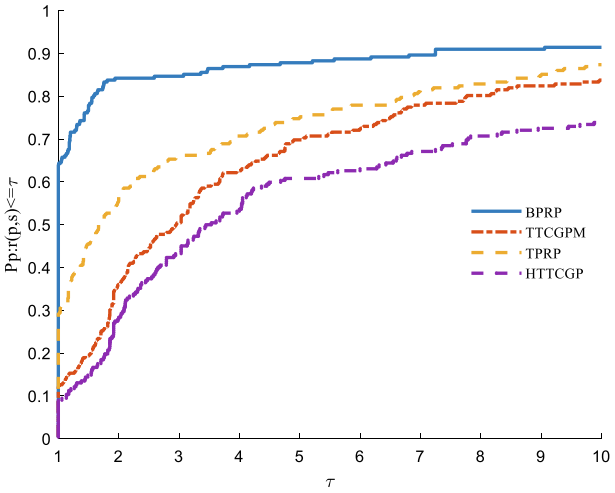

Fig. 2 Performance of algorithms in terms of total of function and gradient evaluations(NFG)

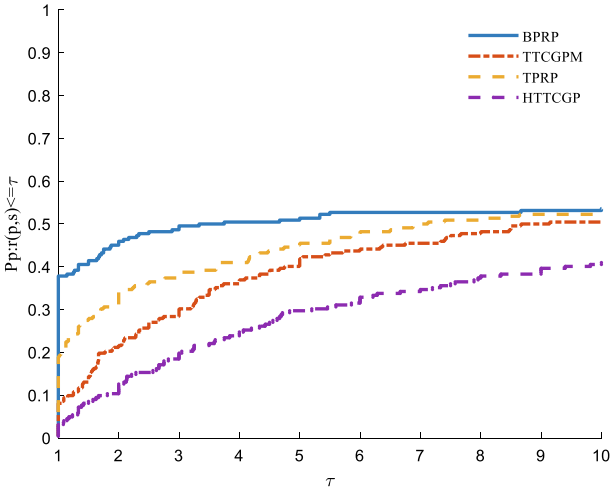

Fig. 3 Performance of algorithms in terms of CPU time

and adding function value information in the search direction does not cause too much computation burden. Figure 3 shows that the BPRP algorithm can solve 40% of the problems first, and the computational efficiency is relatively high. The performance curve include NI, NFG and CPU time, demonstrates the superior performance of the BPRP algorithm over TTCGPM, TPRP, and HTTCGP algorithms. In summary, BPRP constitutes an efficient and robust approach for addressing unconstrained optimization problems.

Table 1 Problem descriptions for large-scaled testing problems

No	Problems
1	Generalization of MAXQ (convex)
2	Chained LQ (convex)
3	Chained CB3 (convex)
4	Chained CB3 2 (convex)
5	Number of active faces (nonconvex)
6	Nonsmooth generalization of Brown function 2 (nonconvex)
7	Chained Mifflin 2 (nonconvex)
8	Chained crescent (nonconvex)
9	Chained crescent 2 (nonconvex)

4.2 Nonsmooth problems

Given significant advantages of CG algorithm in handling large-scale problems, we study BPRP algorithm's efficacy for large-scale non-smooth problems. A comparative analysis is conducted, juxtaposing the BPRP algorithm with the structurally HTTCGP algorithm in [13]. Compared to the BPRP algorithm, the primary distinction of the HTTCGP algorithm is the selection of ω_k and y_k. This is precisely the area where we have made modifications, and we can provide proof of their effectiveness.The test issues listed in Table 1 are taken from [14]. The algorithm stops when $F^a(x_{k-1}, \delta_{k-1}) - F^a(x_k, \delta_k) < 10^{-7}$ is satisfied. The data are listed in Table 2. Parameters were chosen as $c_1 = 100$, $c_2 = 100$ and $t_k = \min\{\tau, \max\{0, 1 - \frac{y_{k-1}^{m\ T} s_{k-1}}{\|y_{k-1}^m\|^2}\}\}$ where $\tau = 0.1$.

Observing data in Table 2, BPRP algorithm has certain competitiveness in the number of iterations and the number of function evaluation, and the final function value is more satisfactory. Additionally, as dimensionality increases, the number of iterations do not exhibit a significant escalation. In summary, it can be conclusively affirmed that the BPRP algorithm proves to be effective in this context.

4.3 The Muskingum model in engineering problems

In solving many practical problems in life, optimization plays a vital role, especially in the field of engineering applications can not be ignored. In order to achieve excellent performance in engineering problems, an excellent optimization algorithm is essential. In the field of hydrologic engineering, Muskingum model is widely used to deal with flood flow problems, and improving the accuracy of model parameters is of great significance for problem solving. Determine Muskingum model's parameters by applying BPRP modified PRP algorithm, compare the performance of this approach with the HTTCGP, TPRP, and TTCGPM algorithms. The Muskingum model, as defined by

Table 2 NI, NF(number of function evaluations) and $f(x)$(eventual iteration) data of BPRP and HTTCGP algorithm

No.	Dim	NI/NF/$f(x)$ BPRP algorithm	HTTCGP algorithm
1	5000	215/4479/6.99559482635975E-06	228/4752/7.43978062640373E-06
	10000	226/4710/6.91298266475338E-06	240/5004/7.08099233299148E-06
	50000	251/5235/6.99643021634188E-06	266/5550/7.69099485781992E-06
	100000	262/5466/6.78867206762538E-06	278/5802/7.19461485144772E-06
2	5000	48/130/8.13861702864239E-07	61/169/1.30490663259508E-06
	10000	63/161/8.22819310067731E-07	76/200/1.63134634834680E-06
	50000	94/226/5.31314778498398E-07	136/324/1.24371489164597E-06
	100000	109/257/5.37093863090017E-07	151/355/1.55717329020106E-06
3	5000	1377/19603/1.85190789424016E+04	1373/19548/1.85231356918901E+04
	10000	1378/19617/3.70563209400204E+04	1374/19562/3.70634647668958E+04
	50000	1379/19631/1.85155957400767E+05	1375/19576/1.85193214890018E+05
	100000	1379/19631/3.70531918897898E+05	1375/19576/3.70599996784097E+05
4	5000	1391/19628/1.85177917384259E+04	1387/19573/1.85218768947581E+04
	10000	1392/19642/3.70538333110851E+04	1388/19587/3.70610251864273E+04
	50000	1393/19656/1.85142664971880E+05	1389/19601/1.85180201829834E+05
	100000	1393/19656/3.70506789384932E+05	1389/19601/3.70575336709152E+05
5	5000	39/207/5.75517379343837E-07	53/263/1.53022587849940E-06
	10000	44/312/8.33396282210117E-07	59/371/1.41216513803395E-06
	50000	58/588/9.11121761967034E-07	74/650/1.25288865888269E-06
	100000	64/696/6.66748598322243E-07	81/779/1.08931015230706E-06

Table 2 continued

No.	Dim	NI/NF/$f(x)$ BPRP algorithm	HTTCGP algorithm
6	5000	63/161/8.21718572790244E-07	76/200/1.63109274420585E-06
	10000	78/192/8.30754012878062E-07	106/262/1.27449071670263E-06
	50000	109/257/5.36443017212815E-07	151/355/1.55550482808494E-06
	100000	124/288/5.42316549366825E-07	181/417/1.21683136347322E-06
7	5000	34/102/-1.24974999918775E+03	47/141/-1.24974999869513E+03
	10000	63/161/-2.49974999917875E+03	76/200/-2.49974999836807E+03
	50000	94/226/-1.24997499994603E+04	136/324/-1.24997499987505E+04
	100000	109/257/-2.49997499995174E+04	151/355/-2.49997499984549E+04
8	5000	48/130/6.04714978758025E-07	61/168/1.25450880550381E-06
	10000	49/133/6.10940157907969E-07	76/199/1.56719798860560E-06
	50000	65/167/7.80061352845252E-07	94/239/1.19509586493649E-06
	100000	66/170/7.88516598015931E-07	95/242/1.49157975437574E-06
9	5000	13/39/1.54461611612255E-07	17/51/2.75184108722292E-07
	10000	13/39/1.54480125025280E-07	17/51/2.76271010735129E-07
	50000	13/39/1.54918712857999E-07	17/51/2.74979877756820E-07
	100000	13/39/1.54949396979909E-07	17/51/2.76018014777612E-07

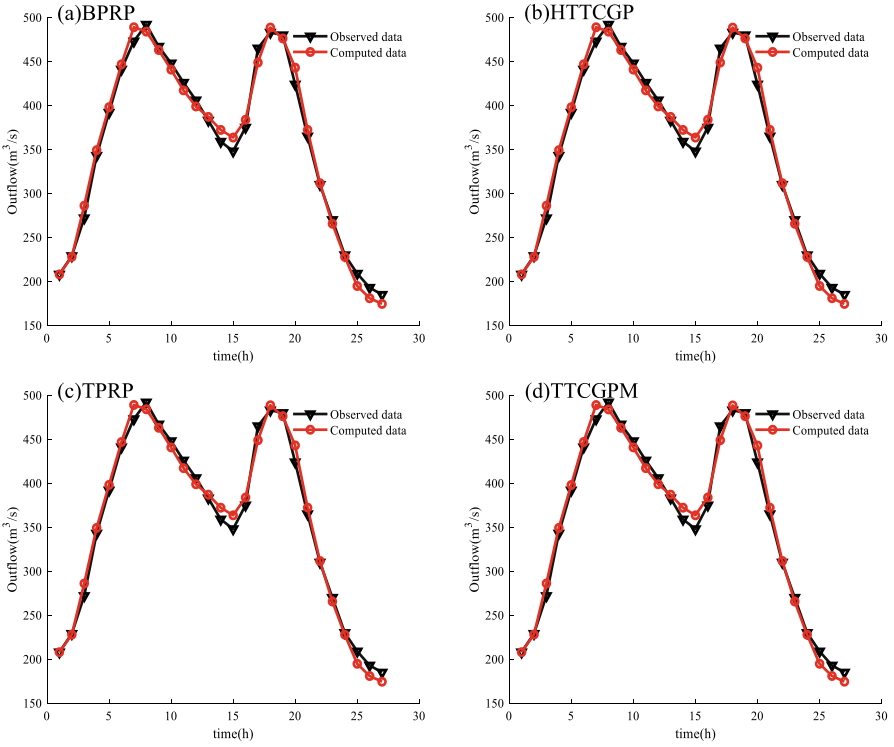

Fig. 4 The Muskingum model in 1960

Ouyang et al. [15], is as follows

$$\begin{aligned}\min f\left(x_1, x_2, x_3\right) = \sum_{i=1}^{n-1} \Bigg(\Bigg(\left(1 - \frac{\Delta t}{6}\right) x_1 \left(x_2 \tilde{I}_{i+1} + (1 - x_2)\, \tilde{Q}_{i+1}\right)^{x_3} \\ - \left(1 - \frac{\Delta t}{6}\right) x_1 \left(x_2 \tilde{I}_i + (1 - x_2)\, \tilde{Q}_i\right)^{x_3} - \frac{\Delta t}{2}\left(\tilde{I}_i - \tilde{Q}_i\right) \\ + \frac{\Delta t}{2}\left(1 - \frac{\Delta t}{3}\right)\left(\tilde{I}_{i+1} - \tilde{Q}_{i+1}\right)\Bigg)^2 .\end{aligned}$$

where the variable n as total time, x_1 as water storage time constant, x_2 is weighting coefficient, and x_3 is supplementary parameter. Δt signifies the length of the calculation period, $\tilde{I}_i$ stands for observed inflow flow, $\tilde{Q}_i$ represents observed outflow flow. Setting initial point as $x = (0, 1, 1)^T$, calculation period Δt is specified as 12 h. The observational data utilized in this experiment are from actual observations of flooding processes along the South Canal of Tianjin Haihe River Basin, spanning the years 1960, 1961, and 1964. Detailed datasets are sourced from [16].

Various algorithms were employed to compute the flows for the years 1960, 1961, and 1964. The calculated results were summarized and juxtaposed with the actual

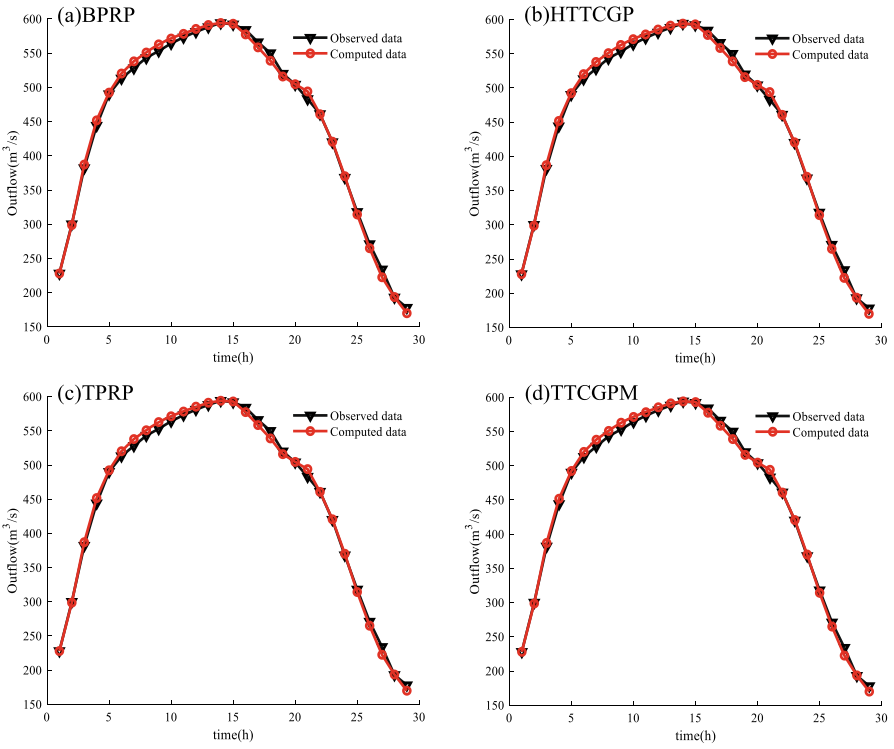

Fig. 5 The Muskingum model in 1961

observed flows, as illustrated in Figs. 4, 5, and 6. The analysis indicates that the data obtained through the BPRP method exhibit no discernible deviation from the observed data, shows comparable performance to other methods. The BPRP method demonstrates no lagging behind in its effectiveness when compared to alternative approaches.

4.4 Image restoration problems

The classic application of optimization problem in real life is to restore the damaged image. In this section, the color image damaged by pulse noise is restored using the BPRP method. ColorCheckerTestImage(1542 × 1024), llama (1314 × 876), car2 (3504 × 2336) and car1 (3504 × 2336) were selected as test images, and 25%, 50% and 75% pulse noise were applied to them. Then the BPRP, TPRP, HTTCGP, and TTCGPM methods were applied to restore the noisy image. The outcomes of this restoration process are showed in Figs. 7, 8, and 9.

To facilitate a more intuitive comparison of the recovery capabilities of various algorithms, the relevant data of the restored images are detailed in Table 3. Peak signal-to-noise ratio (PSNR) is used to evaluate the mean square error between the original image and the restored image, and is one of the commonly used indicators to measure the quality of the restored image. Structural similarity index (SSIM) is

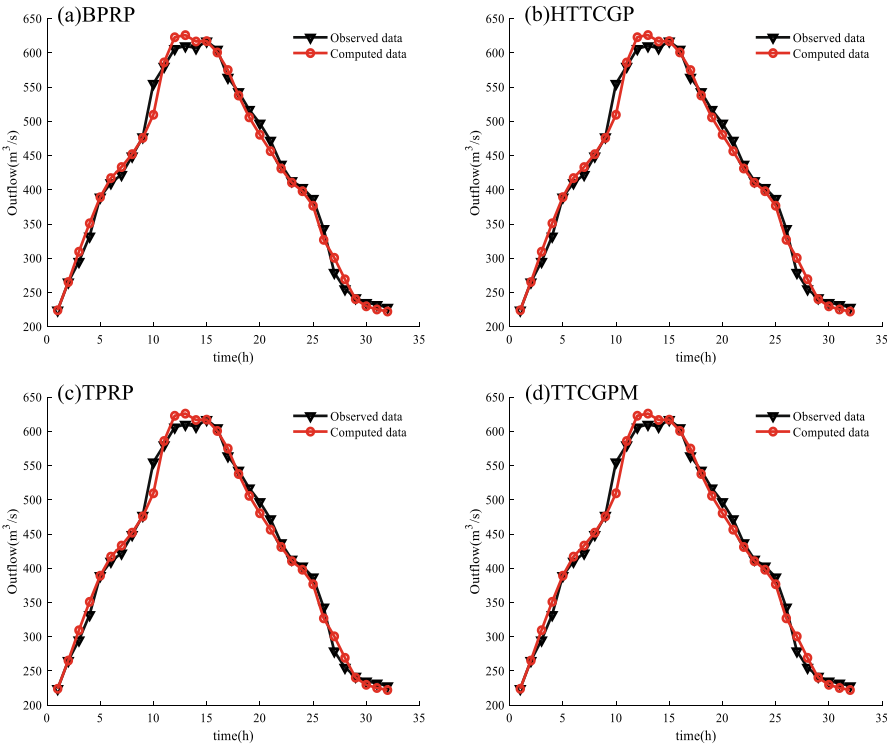

Fig. 6 The Muskingum model in 1964

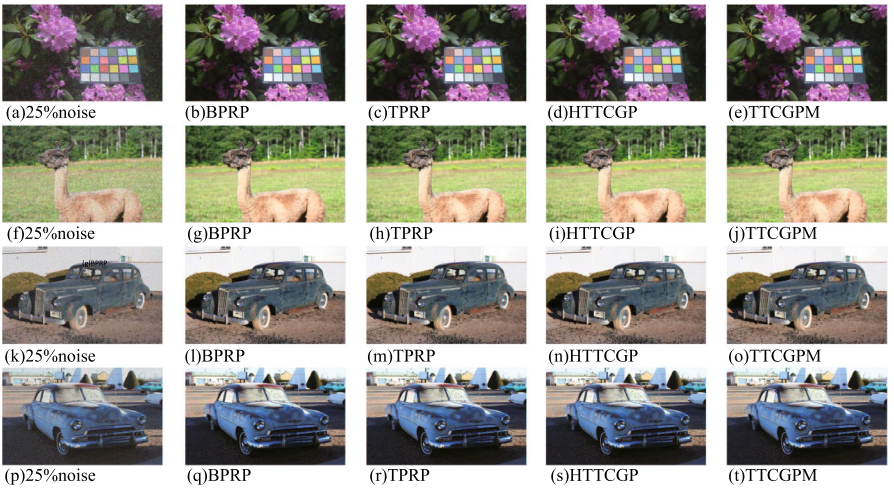

Fig. 7 25% salt-and-pepper noise

Fig. 8 50% salt-and-pepper noise

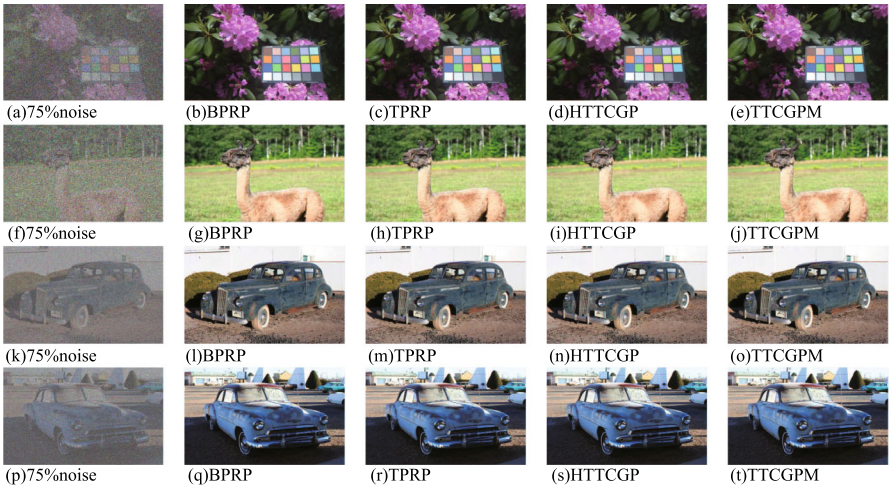

Fig. 9 75% salt-and-pepper noise

a reference index to measure the similarity between images. We consider these two indexes comprehensively to evaluate the image recovery quality. The findings from Table 3 show that: (1) Bprp, TPRP, HTTCGP and TTCGPM methods can all complete image restoration, and the restoration effect is good, SSIM is greater than 0.8, PSNR is similar, and image quality of all four is similar. (2) Among four algorithms, BPRP algorithm has slightly lower CPU time and higher computational efficiency. (3) As the level of noise increases, the quality and efficiency of recovery reduced significantly for all four algorithms, indicating the impact of noise ratio on the overall recovery effectiveness.

Table 3 Peak Signal-to-Noise Ratio (PSNR), Structural Similarity Index (SSIM), and CPU time data of Bprp, TPRP, HTTCGP and TTCGPM algorithms

	colorCheckerTestImage	llama	car2	car1
25% noise				
BPRP	39.2411\0.9929\6.2774	38.0611\0.9939\4.7979	35.7740\0.9833\33.2785	32.0093\0.9739\25.5698
TPRP	39.3186\0.9930\8.7754	38.1096\0.9940\7.0452	35.7892\0.9834\31.1300	31.9970\0.9740\28.7188
TTCGPM	39.3153\0.9930\8.6398	38.1015\0.9940\7.1289	35.7955\0.9834\30.6422	32.0077\0.9739\29.7106
HTTCGP	39.2595\0.9929\8.3829	38.0946\0.9939\7.5439	35.8072\0.9834\30.6598	32.0225\0.9740\29.7049
50% noise				
BPRP	33.7661\0.9761\7.6945	33.2990\0.9821\6.1296	30.6241\0.9480\38.5031	26.9884\0.9220\34.5604
TPRP	33.8085\0.9764\9.9425	33.3411\0.9822\8.1991	30.6302\0.9481\39.8443	27.0066\0.9220\38.4482
TTCGPM	33.7807\0.9763\10.0995	33.3489\0.9820\8.3378	30.6382\0.9480\40.2152	26.9891\0.9220\38.1938
HTTCGP	33.7909\0.9763\10.2519	33.3419\0.9821\8.2281	30.6561\0.9480\39.3975	27.0005\0.9220\38.4587
75% noise				
BPRP	28.8179\0.9326\10.3386	29.2048\0.9559\7.7502	26.2177\0.8691\51.2009	22.8803\0.8180\46.5787
TPRP	28.8429\0.9326\12.4884	29.2301\0.9562\9.6978	26.1998\0.8688\56.0045	22.8889\0.8183\53.4682
TTCGPM	28.8142\0.9325\12.6732	29.2577\0.9562\10.1022	26.2075\0.8686\56.9427	22.8830\0.8182\52.2943
HTTCGP	28.8272\0.9329\12.6431	29.2264\0.9562\9.9373	26.2081\0.8688\55.9412	22.8933\0.8183\51.7816

5 Discussion

The proposed improved CG algorithm achieve sufficient descent and trust region properties, doesn't rely on choice of step size. For unconstrained optimization, the algorithm's global convergence is proven, removing the requirement for Lipschitz continuity conditions. The global convergence of the algorithm is proved for nonsmooth convex problems. The improved algorithm is competitive when compared to other algorithms with comparable structures, according to numerical experiments.

Funding This work is supported by Guangxi Science and Technology base and Talent Project (Grant No. AD22080047), the National Natural Science Foundation of Guangxi Province (Grant No. 2023GXNFSBA 026063), the Innovation Funds of Chinese University (Grant No. 2021BCF03001), and the special foundation for Guangxi Ba Gui Scholars.

Declarations

Conflict of interest The authors declare to have no Conflict of interest.

References

1. Yuan, G., Yang, H., Zhang, M.: Adaptive three-term PRP algorithms without gradient Lipschitz continuity condition for nonconvex functions. Numer. Algorithms **91**(1), 145–160 (2022)
2. Yuan, G., Zhang, M., Zhou, Y.: Adaptive scaling damped BFGS method without gradient Lipschitz continuity. Appl. Math. Lett. **124**, 107634 (2022)
3. Yuan, G., Li, P., Lu, J.: The global convergence of the BFGS method with a modified WWP line search for nonconvex functions. Numer. Algorithms **91**(1), 353–365 (2022)
4. Perry, A.: A class of conjugate gradient algorithms with a two-step variable-metric memory. Technical Report Discussion Paper 269, Center for Mathematical Studies in Economics and Management Sciences, Northwestern University, Evanston, Illinois (1977)
5. Shanno, D.F.: Conjugate gradient methods with inexact searches. Math. Oper. Res. **3**(3), 244–256 (1978)
6. Dai, Y., Kou, C.: A nonlinear conjugate gradient algorithm with an optimal property and an improved Wolfe line search. SIAM J. Optim. **23**(1), 296–320 (2013)
7. Li, M.: A modified Hestense-Stiefel conjugate gradient method close to the memoryless BFGS quasi-Newton method. Optim. Methods Softw. **33**(2), 336–353 (2018)
8. Li, M.: A three term Polak-Ribière-Polyak conjugate gradient method close to the memoryless BFGS quasi-Newton method. J. Ind. Manag. Optim. **16**(1), 245–260 (2018)
9. Yuan, G., Wei, Z.: Convergence analysis of a modified BFGS method on convex minimizations. Comput. Optim. Appl. **47**(2), 237–255 (2010)
10. Conn, A.R., Gould, N.I., Toint, P.L.: vol. 286, pp. 186–195. Society for Industrial and Applied Mathematics (2000)
11. Fukushima, M., Qi, L.: A globally and superlinearly convergent algorithm for nonsmooth convex minimization. SIAM J. Optim. **6**(4), 1106–1120 (1996)
12. Dolan, E.D., Moré, J.J.: Benchmarking optimization software with performance profiles. Math. Program. **91**, 201–213 (2002)
13. Yin, J., Jian, J., Jiang, X., Liu, M., Wang, L.: A hybrid three-term conjugate gradient projection method for constrained nonlinear monotone equations with applications. Numer. Algorithms **88**, 389–418 (2021)
14. Lukšan, L., Vlcek, J.: Test problems for nonsmooth unconstrained and linearly constrained optimization. Technical Report Technical Report No. 798, Institute of Computer Science, Academy of Sciences of the Czech Republic (2000)

15. Ouyang, A., Liu, L.-B., Sheng, Z., Wu, F., et al.: A class of parameter estimation methods for nonlinear Muskingum model using hybrid invasive weed optimization algorithm. Math. Probl. Eng. **2015**, 573894 (2015)
16. Ouyang, A., Tang, Z., Li, K., Sallam, A., Sha, E.: Estimating parameters of Muskingum model using an adaptive hybrid PSO algorithm. Int. J. Pattern Recognit Artif Intell. **28**(1), 1459003 (2014)