Electronic Research Archive

https://www.aimspress.com/journal/era

ERA, 32(11): 6175–6199.
DOI: 10.3934/era.2024287
Received: 19 September 2024
Revised: 01 November 2024
Accepted: 08 November 2024
Published: 15 November 2024

Research article

An efficient modified HS conjugate gradient algorithm in machine learning

Gonglin Yuan[1,2] **and Minjie Huang**[1,2,*]

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

[2] Institute of Data Science and Intelligent Decision Making, Guangxi University, Nanning 530004, China

* **Correspondence:** Email: 2206301012@st.gxu.edu.cn.

Abstract: The Hestenes-Stiefe (HS) conjugate gradient method is very effective in resolving larger-scale sophisticated smoothing optimization tasks due to its low computational requirements and high computational efficiency. Additionally, the algorithm has been employed in practical applications to address image restoration and machine learning issues. In this paper, the authors proposed an improved Hestenes-Stiefe conjugate gradient algorithm having characteristics like: i) The algorithm depicts the decreasing features and trust region properties free of conditionalities. ii) The algorithm satisfies global convergence. iii) The algorithm can be applied to tackle the image restoration problem, monotone nonlinear equations, and machine learning problems. Numerical results revealed that the proffered technique is a competitive method.

Keywords: conjugate gradient; global convergence; image restoration; nonlinear equations; machine learning

1. Introduction

Solving systems of nonlinear equations is an important part of optimization theory and algorithms and is essential to applications in machine learning, artificial intelligence, economic planning, and other important fields. The theoretical study of algorithms for nonlinear systems of equations is an important research topic in the fields of computational mathematics, operations research and optimal control, and numerical algebra. As early as the 1970s, the monographs Orrega [1] and Rheinboldt [2] had done systematic research on nonlinear systems of equations in terms of theory and solutions.

An early and very famous method was the Newtonian method [3]. It has the advantage that the algorithm converges quadratically when the initial point chosen is close to the minima; however, Newton's method does not necessarily converge when the initial point is away from the minima, and Newton's

method requires the computation and storage of Jacobian matrices. So a wide range of researchers have built proposed quasi-Newtonian methods [4, 5]. Such algorithms utilize approximate Jacobian matrices, inheriting the fast convergence of Newton's algorithm. In addition, methods for solving problems with linear equations are the Gaussian-Newton method, Levenberg-Marquardt algorithm, and their various modified forms [6–8]. The above methods necessitate computing and banking Jacobian matrices or approximate Jacobian matrices for each iteration step. The nonlinear conjugate gradient method belongs to the classical computational methods in the first-order optimization methods, which has the characteristics of simple structure, small storage, and low computational complexity, and thus has been widely studied by the optimization community [9–13]. Derivative-free algorithms are one of the popular algorithms for solving large-scale nonlinear systems of equations; they utilize the hyperplane projection technique [14] with a structure of first-order optimization methods, which have an R-linear convergence speed under appropriate conditions. These algorithms have a simple structure, a small amount of storage, a low computational complexity, and they do not require derivative information, thus they are favored by a wide range of researchers.

To ensure the descent of search direction, adjusting search direction structure becomes another important way to study the nonlinear conjugate gradient method. Yuan et al. [15] proposed a further improved weak Wolfe-Powell line-search and proved the method's global convergence in determining average functions given appropriate conditions. [16] proposed the adaptive scaled damped BFGS method (SDBFGS) for solving gradient non-Lipschitz non-convex objective functions. The above approach is attractive because the algorithm has strong self-correcting properties for large eigenvalues. In recent years, under influence from [17], some 1st-order optimization methods exemplified by conjugate gradient (CG) techniques are widely accepted to solve large-scale unconstrained optimization projects and are straightforward and low-memory [9–13]. These extensions, together with the freshly elaborated techniques, are permutations of renowned conjugate gradient algorithms, another key numerical tool for unconstrained optimization [18–21].

The three-term conjugate gradient method [22] is considered to have tantalizing numerical capabilities and good theoretical properties. Yuan [23] presented an adaptive three-term Polak-Ribière-Polyak (PRP) method for non-convex functions and non-Lipschitz continuous functions with gradient. The efficient conjugate gradient algorithm is notorious for requiring both rapid convergence and high precision. [24] proposed a mixed inertia spectral conjugate gradient projection method for solving constrained nonlinear monotone equations, which is superior in solving large-scale equations and recovering blurred images contaminated by Gaussian noise. [25] described two families of self-tuning spectral hybrid DL conjugate gradient methods. The search directions of methods are improved by integrating negative spectral gradients and a final search direction with a convex combinatorial structure. [26] proposed a biased stochastic conjugate gradient (BSCG) algorithm with adaptive step size that combines the stochastic recursive gradient method (SARAH) and the improved Barzilai-Borwein (BB) technique into a typical stochastic gradient algorithm. [27] applied an improved triple conjugate gradient method and linear search technique to machine learning. The same convergence speed as stochastic conjugate gradient algorithm (SCGA) is obtained under weaker conditional assumptions.

The idea of this paper is to combine family weak conjugate gradient methods proposed in [28] with a new parametric formulation of the HS conjugate gradient algorithm proposed in [29]. An efficient HS conjugate gradient method (EHSCG) is proposed and used for image restoration problems and machine learning. Basic characterization of the algorithm is given below:

- The new algorithms have decreasing and trust-region properties requiring no extras.
- They converge globally in well-suited circumstances.
- The new algorithms can solve image restoration, nonlinear monotone equations, and machine learning issues.

In Section 2, we present procedures for solving nonlinear equation models and attest to the related properties. In Section 3, global convergence of the algorithm is proved using the weak Wolfe-Powell line-research condition under normal conditions for non-convex functions. In Section 4, we demonstrate the implementation of the algorithms toward image restoration and machine learning tasks and nonlinear monotone equations.

2. Algorithm

2.1. Motivation of the algorithm

Consider the following nonlinear model:

$$g(x) = 0, \tag{2.1}$$

where $g : \mathfrak{R}^n \to \mathfrak{R}^n$ is a continuously differentiable monotone function, including $x \in \mathfrak{R}^n$. $g(x)$ characterization suggests that the inequality is true:

$$(g(m_1) - g(m_2))^T(m_1 - m_2) \geq 0, \ \forall m_1, \ m_2 \in \mathfrak{R}^n. \tag{2.2}$$

Scholars have come up with a number of interesting theories about this model.

To solve this optimization problem, we typically have iteration $x_{k+1} = x_k + \alpha_k d_k$, which stipulates that symbol x_k signifies the iteration point, x_{k+1} is the following point, α_k is the step length, and d_k is the current direction, which is framed as below:

$$d_k = -g_k + \beta_{k-1} d_{k-1}. \tag{2.3}$$

In [14] covered projection techniques to find large-scale nonlinear equation systems, noted that projection technology is strictly coupled to direction and step size. In particular, hyperplane and projection technologies were leveraged to obtain a formulation for further iteration point:

$$x_{k+1} = \begin{cases} w_k, & \|g(w_k)\| = 0, \\ x_k - \frac{g(w_k)^T(x_k - w_k)g(w_k)}{\|g(w_k)\|^2}, & otherwise, , \end{cases} \tag{2.4}$$

where $w_k = x_k + \alpha_k d_k$.

Furthermore, a linear search solution was presented within [30] to resolve step length α_k in iteration sequencing as follows:

$$-g_{k+1}^T d_k \geq \varpi \alpha_k \|g_{k+1}\|\|d_k\|^2, \tag{2.5}$$

where step length $\alpha_k = \max\left\{\tilde{q}, \tilde{q}\dot{t}, \tilde{q}\dot{t}^2 ...\right\}$, and $\tilde{q} > 0$, $\dot{t} \in (0, 1)$, $\varpi > 0$.

In [28], a modified search direction is followed,

$$d_k^* = -g_k + r_k^\star d_k = -(1 + r_k^\star)g_k + r_k^\star \beta_{k-1} d_{k-1}, \tag{2.6}$$

where d_k is defined in (2.3), and r_{k+1} is designated as follows:

$$r_k^{\star} = \vartheta \frac{\|g_k\|}{\|d_k\|}.$$

The authors show that it can be equated with the traditional HS conjugate gradient method under the conjugation condition. They executed numerous tests that illustrated the superiority of the formulation in tackling large-scale optimization issues.

Yuan et al. [29] updated the parametric formulation with the following HS conjugate gradient algorithm:

$$\hat{d}_k = \begin{cases} -\tau_1 g_k + \frac{g_k^T y_{k-1} d_{k-1} - d_{k-1}^T g_k y_{k-1}}{\max\{\tau_2 \|d_{k-1}\| \|y_{k-1}\|, \tau_3 |d_{k-1}^T y_{k-1}|\}}, & k \geq 2, \\ -\tau_1 g_k, & k = 1, \end{cases} \tag{2.7}$$

where $y_{k-1} = g_k - g_{k-1}$, $\tau_{1,2,3} \geq 0$. The authors prove their methods fulfill an adequate descent condition by converging globally.

Inspired by (2.6) and (2.7), considering both the excellent theoretical and numerical performance of the two algorithms, we obtain the equation below:

$$\begin{aligned} d_k &= -g_k + \vartheta \frac{\|g_k\|}{\|\hat{d}_k\|} \hat{d}_k \\ &= -g_k + \vartheta \frac{\|g_k\|}{\| - \tau_1 g_k + \frac{g_k^T y_{k-1} d_{k-1} - d_{k-1}^T g_k y_{k-1}}{\max\{\tau_2 \|d_{k-1}\| \|y_{k-1}\|, \tau_3 |d_{k-1}^T y_{k-1}|\}} \|} (-\tau_1 g_k + \frac{g_k^T y_{k-1} d_{k-1} - d_{k-1}^T g_k y_{k-1}}{\max\{\tau_2 \|d_{k-1}\| \|y_{k-1}\|, \tau_3 |d_{k-1}^T y_{k-1}|\}}) \\ &= -(1 + \nu_1 \frac{\|g_k\|}{\hbar_k}) g_k + \frac{\|g_k\| (g_k^T y_{k-1} d_{k-1} - d_{k-1}^T g_k y_{k-1})}{\hbar_k \max\{\nu_2 \|d_{k-1}\| \|y_{k-1}\|, \nu_3 |d_{k-1}^T y_{k-1}|\}}, \end{aligned}$$

where $\hbar_k = \sqrt{\tau_1^2 \|g_k\|^2 + \frac{(|g_k^T y_{k-1}| \|d_{k-1}\| - |d_{k-1}^T g_k| \|y_{k-1}\|)^2}{\max\{\tau_2 \|d_{k-1}\| \|y_{k-1}\|, \tau_3 |d_{k-1}^T y_{k-1}|\}^2}}$.

On account of the above derivation, eventually we acquire the improved d_{k+1} formulation of this paper:

$$d_{k+1} = \begin{cases} -(1 + \nu_1 \frac{\|g_{k+1}\|}{\hbar_k}) g_{k+1} + \frac{\|g_{k+1}\| (g_{k+1}^T y_k d_k - d_k^T g_{k+1} y_k)}{\hbar_k \max\{\nu_2 \|d_k\| \|y_k\|, \nu_3 |d_k^T y_k|\}}, & k \geq 1, \\ -(1 + \nu_1 \frac{\|g_{k+1}\|}{\hbar_k}) g_{k+1}, & k = 0, \end{cases} \tag{2.8}$$

where $\hbar_{k+1} = \sqrt{\tau_1^2 \|g_{k+1}\|^2 + \frac{(|g_{k+1}^T y_k| \|d_k\| - |d_k^T g_{k+1}| \|y_k\|)^2}{\max\{\tau_2 \|d_k\| \|y_k\|, \tau_3 |d_k^T y_k|\}^2}}$, $y_k = g_{k+1} - g_k$, $\nu_1 = \vartheta \tau_1$, $\nu_2 = \frac{\tau_2}{\vartheta}$, $\nu_3 = \frac{\tau_3}{\vartheta}$, and $\vartheta > 0$, $\nu_i > 0$, and $\tau_i > 0, i = 1, 2, 3$. Based on (2.8), combined with line search (2.5), the sufficient descent characterization and trust region of the algorithm are provided. Equally the global convergence of the algorithm can be certified. Meanwhile, numerical results also prove the effectiveness and feasibility of the algorithm.

2.2. Convergence analysis

Assumption 2.1. *(i) The problem's solution set is not* $\varnothing$*. (ii) Gradient* $g(x)$ *is Lipschitz continuous, and this implies that the following inequality holds:* $\exists L > 0$*, s.t.*

$$\|g(m_1) - g(m_2)\| \leq L \|m_1 - m_2\|, \ \forall m_1, \ m_2 \in \Re^n. \tag{2.9}$$

Algorithm 1 Efficient HS conjugate gradient method (EHSCG)

1: Recognize an initial point, $x_0 \in \Re^n$; constants ϖ, ϑ, $\tilde{q} > 0$; i, $\varepsilon \in (0, 1)$, $\nu_{1,2,3} > 0$. Let $k = 1$.
2: If $\|g_k\| \leq \varepsilon$, stop; otherwise, calculate d_k based on (2.8).
3: Selecting the right step size α_k on the basis of (2.5).
4: Reset the new point to be $w_k = x_k + \alpha_k d_k$.
5: If $\|g_k\| \leq \varepsilon$, stop, set $x_{k+1} = w_k$. Or else, construct the iteration point x_k on the basis of $x_{k+1} = x_k - \frac{g(w_k)^T(x_k - w_k)g(w_k)}{\|g(w_k)\|^2}$.
6: Let $k = k + 1$, visit 2.

This assumption implies $\exists\, \omega \in \mathbb{C}$ *s.t.*

$$\|g_k\| \leq \omega. \tag{2.10}$$

Theorem 2.1. *If d_k satisfies Eq (2.8), then*
(1) proposed descent:

$$g_{k+1}^T d_{k+1} \leq -(1 - \vartheta)\|g_{k+1}\|^2, \tag{2.11}$$

and (2) trust domain:

$$\|d_{k+1}\| \leq (1 + \vartheta)\|g_{k+1}\|. \tag{2.12}$$

Proof. A remarkably similar proof procedure has been granted in Yuan's work in [28] and will not be reiterated for here. □

Lemma 2.2. *If Assumption 2.1 holds and the objective function monotones, the iterative series $\{x_k\}$ is yielded by Algorithm 1, and point $x^\star$ satisfies the condition $g(x^\star) = 0$. Taking it a step further, if the series x_k is infinite, then*

$$\sum_{k=0}^{\infty} \|x_k - x_{k-1}\|^2 \leq \infty. \tag{2.13}$$

Proof. A detailed proof procedure of it is available in [14]. □

Theorem 2.3. *If Assumption 2.1 holds, Algorithm 1 produces a finite series of step iterations $\{\alpha_k\}$ during the iteration from an active point to the newer one.*

Proof. This conclusion is supported by contradiction. Assume the indication set $\varphi = N \cup \{0\}$, take any $k \in \varphi$, and consider the iteration of x_k. Assume that the step size satisfying the line search does not occur; that is, there occurs a step size such that $\alpha^\star = \tilde{q}i^a$ satisfies:

$$-\mathring{g}^T d_k \leq \varpi \alpha^\star \|\mathring{g}\| \|d_k\|^2, \tag{2.14}$$

where $\mathring{g} = g(x_k + \alpha^\star d_k)$. Referring to the previous (2.9) and (2.14), we have

$$\begin{aligned} \|g_k\|^2 &= -\frac{1}{1 + \vartheta \frac{\|g_k\|}{\hbar_k}} g_k^T d_k \\ &= \frac{1}{1 + \vartheta \frac{\|g_k\|}{\hbar_k}} ((\mathring{g} - g_k)^T d_k - \mathring{g}^T d_k) \end{aligned}$$

$$
\begin{aligned}
&\leq \frac{1}{1+\vartheta\frac{\|g_k\|}{\hbar_{k-1}}}(\|\mathring{g}-g_k\|\|d_k\|+\varpi\alpha^\star\|\mathring{g}\|\|d_k\|^2)\\
&\leq \frac{1}{1+\vartheta\frac{\|g_k\|}{\hbar_{k-1}}}(L\|x_k+\alpha^\star d_k-x_k\|\|d_k\|+\varpi\alpha^\star\|\mathring{g}\|\|d_k\|^2)\\
&\leq \frac{1}{1+\vartheta\frac{\|g_k\|}{\hbar_k}}(L\alpha^\star\|d_k\|^2+\varpi\alpha^\star\|\mathring{g}\|\|d_k\|^2)\\
&\leq \frac{1}{1+\vartheta\frac{\|g_k\|}{\hbar_k}}\alpha^\star(L+\varpi\|\mathring{g}\|)\|d_k\|^2.
\end{aligned}
$$

By (2.9) and (2.10), then

$$
\begin{aligned}
\|\mathring{g}\| &\leq \|\mathring{g}-g_k\|+\|g_k\|\\
&\leq L\|x_k+\alpha^\star d_k-x_k\|+\omega\\
&= L\alpha^\star\|d_k\|+\omega\\
&= L\alpha^\star(1+\vartheta)\|g_k\|+\omega\\
&\leq (L\alpha^\star(1+\vartheta)+1)\omega.
\end{aligned}
$$

Further we obtain that

$$
\begin{aligned}
\alpha^\star &\geq \frac{(1+\vartheta\frac{\|g_k\|}{\hbar_k})\|g_k\|^2}{(L+\varpi\|g(x_k+\alpha^\star d_k)\|)\|d_k\|^2}\\
&\geq \frac{(1+\vartheta\frac{\|g_k\|}{\hbar_k})\|g_k\|^2}{(L+\varpi(L\tilde{q}(1+\vartheta)+1)\omega)\|d_k\|^2}\\
&\geq \frac{2+\vartheta}{(L+\varpi(L\tilde{q}(1+\vartheta)+1)\omega)(1+\vartheta)^3}>0.
\end{aligned}
$$

The above inequality shows that the step size $\alpha^\star$ is bounded. □

Theorem 2.4. *If Assumption 2.1 stands, Algorithm 1 yields* $\{\alpha_k\}$, $\{d_k\}$, $\{x_k\}$, *and* $\{g_k\}$, *and then* $\liminf_{k\to\infty}\|g_k\|=0$.

Proof. This theorem is still proved by the converse method. We denote the indicator set $\varphi = N\cup\{0\}$, and set $k\in\varphi$. According to Assumption 2.1, assume that $\exists\,\varrho,\ n_0>0$ *s.t.*

$$
\|g_k\|\geq\varrho \text{ and } \forall k\geq n_0, \tag{2.15}
$$

where ϱ is a constant and n_0 is an index. According to (2.11) and (2.10), we have

$$
\|d_{k+1}\|\leq(1+\vartheta)\|g_{k+1}\|\leq(1+\vartheta)\varrho.
$$

Consider the Eq (2.11) with (2.12), and we have

$$
\|g_k\|\|d_k\|\geq g_k^T d_k=-(1+\frac{\|g_k\|}{\hbar_k})\|g_k\|^2,
$$

$$\|d_k\| \geq -(1+\vartheta\frac{\varrho}{\|d_k\|})\varrho,$$

$$\|d_k\|^2 - \varrho\|d_k\| \geq \vartheta\varrho^2,$$

$$(\|d_k\| - \frac{\varrho}{2})^2 \geq (\vartheta + \frac{1}{4})\varrho^2,$$

$$\|d_k\| \geq \sqrt{\vartheta + \frac{1}{4}}\varrho + \frac{\varrho}{2},$$

$$\sqrt{\vartheta + \frac{1}{4}}\varrho + \frac{\varrho}{2} \leq \|d_k\| \leq (1+\vartheta)\varrho.$$

It can be concluded that direction d_{k+1} is bounded, we suppose that $\lim_{k\to\infty} d_k = d^\star$, and follow (2.13) with the Theorem 2.3 iteration points $\lim_{k\to\infty} x_k = x^\star$. Step size α_k is bounded, and we have

$$\sum_{k=0}^{\infty} \|x_{k+1} - x_k\|^2 = \sum_{k=0}^{\infty} \|\alpha_k d_k\|^2 < \infty, \tag{2.16}$$

$$\|\alpha_k d_k\| \to 0,$$

$$\alpha_k \to 0.$$

We obtain that the following inequality,

$$-g_{k+1}^T d_k \leq \varpi\alpha_k\|g_{k+1}\|\|d_k\|^2.$$

Both parts of the given inequality take the limit, the one has

$$-(g^\star)^T d^\star \leq 0.$$

Let k tend to infinity, $g_{k+1}^T d_{k+1} = -(1+\vartheta\frac{\|g_{k+1}\|}{\hbar_{k+1}})\|g_{k+1}\|^2$, and each side of

$$(g^\star)^T d^\star \leq 0,$$

which implies

$$\|g^\star\| = 0,$$

but this contradicts (2.15). Thus, the conclusion of the theorem holds. □

3. Unconstrained optimization problems

We apply (2.8) to the massive unconstrained optimization problem. Consider the issues below:

$$\min_{x\in\Re^n} f(x), \tag{3.1}$$

where $f : \Re^n \to \Re$ is a continuously differentiable function. Set $\nabla f(x) = F(x)$.

Motivated by the description in Section 2, the following formula for d_k is proposed:

$$d_k = \begin{cases} -(1+\nu_1\frac{\|F_k\|}{\hbar_k})F_k + \frac{\|F_k\|(F_k^T\hat{y}_{k-1}d_{k-1} - d_{k-1}^T F_k\hat{y}_{k-1})}{\hbar_k \max\{\nu_2\|d_{k-1}\|\|\hat{y}_{k-1}\|, \nu_3|d_{k-1}^T\hat{y}_{k-1}\|\}}, & k \geq 2, \\ -(1+\nu_1\frac{\|F_k\|}{\hbar_k})F_k, & k = 1, \end{cases} \tag{3.2}$$

where $\hbar_k = \sqrt{\tau_1^2\|F_k\|^2 + \frac{(|F_k^T\hat{y}_{k-1}|\|d_{k-1}\| - |d_{k-1}^T F_k\|\hat{y}_{k-1}\|)^2}{\max\{\tau_2\|d_{k-1}\|\|\hat{y}_{k-1}\|, \tau_3|d_{k-1}^T\hat{y}_{k-1}\|\}^2}}$, $\hat{y}_k = F_{k+1} - F_k$, $\nu_1 = \vartheta\tau_1$, $\nu_2 = \frac{\tau_2}{\vartheta}$, $\nu_3 = \frac{\tau_3}{\vartheta}$, $\vartheta > 0$, and $\tau_i > 0, i = 1, 2, 3$. Analogous to the targeted algorithm in Section 2, d_k bears the following traits:

$$F_k^T d_k \leq -(1-\vartheta)\|F_k\|^2, \tag{3.3}$$

and

$$\|d_k\| \leq (\vartheta + 1)\|F_k\|, \tag{3.4}$$

where $\vartheta > 0$ is a constant. The proofs of the above traits have been given in Section 2 and will not be repeated in this section.

This section presents the algorithm and global convergence thesis.

Algorithm 2

1: Recognize point $x_0 \in \mathfrak{R}^n$; constants $Eps \in (0, 1)$, $0 < \zeta < \frac{1}{2}, \zeta < \xi < 1$; ϑ, $\nu_{1,2,3} > 0$. Let $k = 1$.
2: If $\|F_k\| \leq Eps$, stop; or calculate d_k based on (3.2).
3: Select the step-size α_k based on

$$f_{k+1} \leq f_k + \zeta\alpha_k F_k^T d_k, \tag{3.5}$$

and

$$F_{k+1}^T \geq \xi F_k^T d_k. \tag{3.6}$$

4: Reset the new point to be $x_{k+1} = x_k + \alpha_k d_k$.
5: Let $k := k + 1$, and go to 2.

Assumption 3.1. *(i) The level set $\Theta = \{x|f(x) \leq f(x_0)\}$ bounds.*
(ii) $f(x) \in C^2$ bounds below and Lipschitz continues, which implicates that there exists constant L higher than zero so that

$$\|F(m_1) - F(m_2)\| \leq L\|m_1 - m_2\|,\ m_1,\ m_2 \in \mathfrak{R}^n. \tag{3.7}$$

Theorem 3.1. *If Assumption 3.1 holds, Algorithm 2 creates $\{x_k\}$, $\{\alpha_k\}$, $\{d_k\}$, and $\{F_k\}$, and then*

$$\lim_{k\to\infty}\|F_k\| = 0. \tag{3.8}$$

Proof. We will establish this through counterfactuals. Assuming that the original theorem does not hold, and we will have the following conclusion: an existence of a constant ε^* larger than zero, $\forall k \geq k^* > 0$ *s.t.*

$$\|F_k\| \geq \varepsilon^*.$$

By the decreasing trait (3.3) and line search method (3.5), then

$$f_{k+1} - f_k \leq \zeta\alpha_k F_k^T d_k \leq -\zeta(1-\vartheta)\alpha_k\|F_k\|^2.$$

Consequently, let k range from 0 to ∞, and then

$$\sum_{k=0}^{\infty}(f_{k+1}-f_k)=f_\infty-f_0=\sum_{k=0}^{\infty}(-\zeta(1-\vartheta)\alpha_k\|F_k\|^2)<\infty, \tag{3.9}$$

where the inequality on the right side results from Assumption 3.1(ii). Significantly, Eq (3.9) expresses that

$$\lim_{k\to\infty}-\zeta(1-\vartheta)\alpha_k\|F_k\|^2=0.$$

Through the linear search method (3.6) with Assumption 3.1, then

$$\begin{aligned}(F_k-F_{k-1})^T d_{k-1}&\geq(\xi-1)F_{k-1}^T d_{k-1}\\&\geq-(\xi-1)(1-\vartheta)\|F_{k-1}\|^2,\end{aligned}$$

$$\begin{aligned}\|F_k-F_{k-1}\|\|d_{k-1}\|&\leq L\|\alpha_{k-1}d_{k-1}\|\|d_{k-1}\|\\&\leq L(1+\vartheta)|\alpha_{k-1}|\|F_k\|^2.\end{aligned}$$

In conclusion, we have the fact that

$$\begin{aligned}\alpha_{k-1}&\geq\frac{(1-\xi-\vartheta+\xi\vartheta)\|F_{k-1}\|^2}{L(1+\vartheta)\|F_k\|^2}\\&\geq\frac{1-\xi-\vartheta+\xi\vartheta}{L+L\vartheta}>0,\end{aligned}$$

where these two inequalities are given from (2.12). Hence we obtain (3.8). The proof is finalized. □

4. Numerical results

All methods are encoded in MATLAB R2021b running on a PC with a Windows 10 operating system with an Intel(R) Core(TM) i5-8265U CPU @ 1.60GHz 1.80 GHz.

4.1. Monotone linear equations

We compare Algorithm 1 with the modified HS method [29], three-term PRP method [31], and traditional PRP method [21, 32]. The termination conditions for each of the eleven test questions are

$$\|g_k\|\leq 10^{-5}.$$

The parameters in Algorithm 1 are chosen as: $\varpi=0.01$, $\vartheta=0.8$, $\tilde{q}=1$; $\dot{t}=0.5$, $\nu_1=1.8$, $\nu_2=3000$, and $\nu_3=1000$. We list those problems from [33] here to preserve the neutrality of this paper. The corresponding problems with a matching initial point x_0 are tabulated here:

$$g(x)=(g_1(x),g_2(x),...,g_s(x))^T.$$

Problem 4.1. *Exponential function 1:*

$$g_1(x)=e^{x_1-1}-1, g_t(x)=t(e^{x_t-1}-x_t), t=1,2...,s.$$

point $x_0=(\frac{s}{s-1},\frac{s}{s-1}...,\frac{s}{s-1},)^T$.

Problem 4.2. *Singular function:*

$$g_1(x) = \frac{1}{3}x_1^3 + \frac{1}{2}x_2^2, g_t(x) = -\frac{1}{2}x_t^2 + \frac{t}{3}x_t^3 + \frac{1}{2}x_{t+1}^2, g_s(x) = -\frac{1}{2}x_s^2 + \frac{s}{3}x_s^3, t = 1, 2..., s.$$

point $x_0 = (1, 1..., 1)^T$.

Problem 4.3.

$$g_t(x) = ln(x_t + 1) - \frac{x_t}{s}, t = 1, 2..., s.$$

point $x_0 = (1, 1..., 1)^T$.

Problem 4.4. *Trigexp function:*

$$g_1(x) = 3x_1^3 + 2x_2 - 5 + sin(x_1 - x_2)sin(x_1 + x_2),$$

$$g_t(x) = -x_{t-1}e^{x_{t-1}-x_t} + x_t(4 + 3x_t^2) + 2x_{t-1} + sin(x_t - x_{t-1})sin(x_t + x_{t-1}) - 8,$$

$$g_s(x) = -x_s e^{x_{s-1}-x_s} + 4x_s - 3, t = 1, 2..., s.$$

point $x_0 = (0, 0..., 0)^T$.

Problem 4.5.

$$g_t(x) = e^{x_t} - 1, t = 1, 2..., s.$$

point $x_0 = (\frac{1}{s}, \frac{2}{s}..., \frac{s}{s})^T$.

Problem 4.6. *Penalty I function:*

$$g_t(x) = \sqrt{10^{-5}}(x_t - 1), t = 1, 2..., s - 1.$$

$$g_s(x) = \frac{1}{4s}(x_1^2 + x_2^2 + ... + x_s^2) - \frac{1}{4}.$$

point $x_0 = (\frac{1}{3}, \frac{1}{3}..., \frac{1}{3})^T$.

Problem 4.7. *Variable dimensioned function:*

$$g_t(x) = x_t - 1, t = 1, 2..., s - 1.$$

$$g_{s-1}(x) = (x_1 - 1) + 2(x_2 - 1) + 3(x_3 - 1) + ... + (s - 2)(x_{s-2} - 1),$$

$$g_s(x) = ((x_1 - 1) + 2(x_2 - 1) + 3(x_3 - 1) + ... + (s - 2)(x_{s-2} - 1))^2.$$

point $x_0 = (1 - \frac{1}{s}, 1 - \frac{2}{s}..., 1 - \frac{s}{s})^T$.

Problem 4.8. *Tridiagonal system:*

$$g_1(x) = 4(x_1 - x_2^2),$$

$$g_t(x) = 8x_t(x_t^2 - x_{t-1}) - 2(1 - x_t) + 4(x_t - x_{t+1}^2), t = 2, 3..., s - 1,$$

$$g_s(x) = 8x_s(x_s^2 - x_{s-1}) - 2(1 - x_s).$$

point $x_0 = (12, 12..., 12)^T$.

Problem 4.9. *Five-diagonal system:*

$$g_1(x) = 4(x_1 - x_2^2) + x_2 - x_3^2,$$

$$g_2(x) = 8x_2(x_2^2 - x_1) - 2(1 - x_2) + 4(x_2 - x_3^2) + x_3 - x_4^2,$$

$$g_t(x) = 8x_t(x_t^2 - x_{t-1}) - 2(1 - x_t) + 4(x_t - x_{t+1}^2) + x_{t-1}^2 - x_{t-2} + x_{t+1} - x_{t+2}^2, t = 3, 4..., s - 2,$$

$$g_{s-1}(x) = 8x_{s-1}(x_{s-1}^2 - x_{s-2}) - 2(1 - x_{s-1}) + 4(x_{s-1} - x_s^2) + x_{s-2}^2 - x_{s-3},$$

$$g_s(x) = 8x_s(x_s^2 - x_{s-1}) - 2(1 - x_s) + x_{s-1}^2 - x_{s-2}.$$

point $x_0 = (-2, -2..., -2)^T$.

Problem 4.10. *Seven-diagonal system:*

$$g_1(x) = 4(x_1 - x_2^2) + x_2 - x_3^2 + x_3 - x_4^2,$$

$$g_2(x) = 8x_2(x_2^2 - x_1) - 2(1 - x_2) + 4(x_2 - x_3^2) + x_1^2 + x_3 - x_4^2 + x_4 - x_5^2,$$

$$g_3(x) = 8x_3(x_3^2 - x_2) - 2(1 - x_3) + 4(x_3 - x_4^2) + x_2^2 - x_1 + x_4 - x_5^2 + x_1^2 + x_5 - x_6^2,$$

$$\begin{aligned} g_t(x) = {} & 8x_t(x_t^2 - x_{t-1}) - 2(1 - x_t) + 4(x_t - x_{t+1}^2) + x_{t-1}^2 - x_{t-2} + x_{t+1} - x_{t+2}^2 + x_{t-2}^2 \\ & + x_{t+2} - x_{t-3} - x_{t+3}^2, t = 4, 5..., s - 3, \end{aligned}$$

$$g_{s-1}(x) = 8x_{s-1}(x_{s-1}^2 - x_{s-2}) - 2(1 - x_{s-1}) + 4(x_{s-1} - x_s^2) + x_{s-2}^2 - x_{s-3} + x_s + x_{s-3}^2 - x_{s-4},$$

$$g_s(x) = 8x_s(x_s^2 - x_{s-1}) - 2(1 - x_s) + x_{s-1}^2 - x_{s-2} + x_{s-2}^2 - x_{s-3}.$$

point $x_0 = (-3, -3..., -3)^T$.

Problem 4.11. *Troesch problem:*

$$g_1(x) = 2x_1 + 10(\frac{1}{s+1})^2 sin(10\frac{1}{s+1}x_1) - x_2,$$

$$g_t(x) = 2x_t + 10(\frac{1}{s+1})^2 sin(10\frac{1}{s+1}x_t) - x_{t-1} - x_{t+1}, t = 2, 3..., s - 1,$$

$$g_s(x) = 2x_s + 10(\frac{1}{s+1})^2 sin(10(\frac{1}{s+1})^2 x_s) - x_{s-1}.$$

point $x_0 = (0, 0..., 0)^T$.

Table 1. Numerical results of the considered methods.

		Alogtithmn 1	Modified HS	Three-term PRP	PRP
NO	Dim	NI\NG\CPU\$|g_k|$	NI\NG\CPU\$|g_k|$	NI\NG\CPU\$|g_k|$	NI\NG\CPU\$|g_k|$
1	1000	293/451/0.140625/9.941213e-06	395/518/0.421875/9.960606e-06	176/177/0.078125/9.999271e-06	187/188/0.062500/9.923051e-06
1	5000	165/257/0.921875/9.904049e-06	199/270/1.359375/9.989538e-06	105/106/0.609375/9.978384e-06	109/110/0.421875/9.912454e-06
1	10,000	126/198/1.781250/9.824845e-06	155/211/2.000000/9.923827e-06	85/86/1.125000/9.994038e-06	86/87/1.156250/9.942374e-06
1	50,000	55/92/1.875000/9.742663e-06	82/113/2.515625/9.859854e-06	47/48/1.000000/9.970994e-06	50/51/1.812500/9.812461e-06
2	1000	10251/12764/6.437500/9.988723e-06	13248/13441/7.421875/9.984307e-06	24134/29845/14.515625/9.996621e-06	23798/29299/14.640625/9.996136e-06
2	5000	14093/17753/114.421875/9.991092e-06	14934/15561/110.281250/9.991112e-06	24335/42282/243.843750/9.994119e-06	23669/41392/236.50,0000/9.999623e-06
2	10,000	52760/199219/1710.968750/9.997897e-06	15418/16372/218.046875/9.996137e-06	20300/49021/466.765625/9.996696e-06	24989/53363/558.859375/9.990332e-06
2	50,000	14291/40231/991.046875/9.992350e-06	12791/15588/472.906250/9.997958e-06	22215/103750/2252.015625/9.997605e-06	22660/103815/2278.734375/9.998269e-06
3	1000	8/16/0.015625/2.973142e-06	29/110/0.031250/9.327860e-06	32/2250/0.375000/1.017432e-08	41/2195/0.343750/3.078878e-06
3	5000	8/16/0.046875/6.327004e-06	58/300/0.734375/8.380619e-06	68/6691/16.625000/2.462019e-07	76/6599/16.578125/2.149830e-06
3	10,000	8/16/0.125000/8.892321e-06	80/460/2.968750/8.076240e-06	96/10516/62.625000/1.147079e-09	106/10419/62.609375/1.607070e-06
3	50,000	10/22/0.640625/3.745708e-06	166/1152/15.50,0000/6.854968e-06	211/28900/385.796875/5.328693e-08	222/28783/388.671875/2.820361e-06
4	1000	21/126/0.046875/6.619060e-06	76/484/0.125000/9.029869e-06	4107/375958/66.140625/5.376997e-06	263/25956/4.50,0000/9.461107e-06
4	5000	21/126/0.328125/6.579547e-06	101/752/1.843750/6.951950e-06	2534/236972/613.250,000/9.063035e-06	286/32183/82.093750/9.754050e-06
4	10,000	21/126/0.875000/6.570692e-06	121/983/6.312500/8.222816e-06	2505/237950/1454.921875/6.132250e-06	319/38772/230.062500/9.991339e-06
4	50,000	21/127/1.796875/8.942224e-06	212/2078/27.140625/8.051826e-06	3857/378376/5362.843750/7.595834e-06	441/67359/953.593750/9.805361e-06
5	1000	10/23/0.015625/1.345156e-06	20/71/0.015625/7.757674e-06	69/1450/0.250,000/9.446592e-06	27/1188/0.218750/6.659137e-06
5	5000	10/23/0.140625/2.987337e-06	37/176/0.437500/6.260531e-06	81/3869/9.406250/8.030132e-06	48/3627/8.281250/1.916037e-06
5	10,000	10/23/0.343750/4.221010e-06	48/252/1.828125/6.741102e-06	91/5980/36.359375/8.876877e-06	64/5747/32.640625/2.819940e-06
5	50,000	11/26/0.421875/1.273572e-06	100/668/8.031250/7.429674e-06	139/16176/210.640625/9.126806e-06	131/15985/206.812500/9.985455e-06
6	1000	5713/5714/1.671875/9.994632e-06	6856/6857/2.031250/9.996719e-06	10285/10286/2.921875/9.999930e-06	10295/10296/2.781250/9.999854e-06
6	5000	17431/17432/91.203125/9.999935e-06	20918/20919/96.125000/9.999824e-06	31379/31380/151.609375/9.999607e-06	31389/31390/129.140625/9.999784e-06
6	10,000	27627/27628/324.546875/9.999054e-06	33153/33154/369.937500/9.999230e-06	49731/49732/579.281250/9.999510e-06	49741/49742/613.953125/9.999691e-06
6	50,000	72200/72201/1871.421875/9.999723e-06	86640/86641/1789.953125/9.999971e-06	129963/129964/3045.625000/9.999862e-06	129974/129975/3245.187500/9.999835e-06
7	1000	1/2/0.000000/0.000000e+00	1/2/0.000000/0.000000e+00	1/2/0.000000/0.000000e+00	1/2/0.000000/0.000000e+00
7	5000	1/2/0.000000/0.000000e+00	1/2/0.000000/0.000000e+00	1/2/0.000000/0.000000e+00	1/2/0.000000/0.000000e+00
7	10,000	1/2/0.000000/0.000000e+00	1/2/0.062500/0.000000e+00	1/2/0.031250/0.000000e+00	1/2/0.000000/0.000000e+00
7	50,000	1/2/0.031250/0.000000e+00	1/2/0.015625/0.000000e+00	1/2/0.031250/0.000000e+00	1/2/0.046875/0.000000e+00
8	1000	7517/52847/9.609375/9.705240e-06	6256/51420/8.187500/9.885174e-06	3867/637967/91.906250/8.900084e-06	3993/663519/92.890625/9.960198e-06
8	5000	7379/51894/130.171875/9.832652e-06	6535/56778/130.609375/9.978493e-06	4801/844439/1665.171875/8.187114e-06	4368/814256/1623.625000/9.935312e-06
8	10,000	8284/58431/381.203125/9.808342e-06	6762/61024/383.843750/9.976333e-06	4595/896457/5102.390625/8.054508e-06	4581/921975/5151.968750/9.842915e-06
8	50,000	7311/51836/708.437500/9.997653e-06	7554/78279/862.50,0000/9.988038e-06	10463/1986698/21108.781250/9.966481e-06	5525/1401847/15546.796875/9.902653e-06
9	1000	1315/8569/1.50,0000/9.980071e-06	1724/11004/2.187500/9.995043e-06	514/65030/9.437500/9.645362e-06	332/43592/6.50,0000/9.394169e-06
9	5000	1327/8684/21.875000/9.904454e-06	1465/9883/24.593750/9.959200e-06	574/84298/172.421875/9.690180e-06	421/66124/138.812500/9.520384e-06
9	10,000	1335/8771/57.531250/9.914197e-06	1537/10704/71.359375/9.886892e-06	2870/356611/2051.125000/9.956931e-06	493/85160/479.187500/9.995629e-06
9	50,000	1426/9690/137.171875/9.912140e-06	1792/14088/163.593750/9.996782e-06	988/194924/2254.156250/9.358423e-06	884/182395/2254.734375/9.294612e-06
10	1000	5788/46645/7.625000/9.518479e-06	345/3005/0.671875/8.406790e-06	422/70159/11.015625/4.510865e-06	425/70205/11.125000/9.836083e-06
10	5000	5669/45772/118.531250/9.521868e-06	608/5639/14.140625/4.101871e-06	597/110926/237.421875/6.498110e-06	500/97170/209.906250/8.589743e-06
10	10,000	5661/45800/298.265625/9.734609e-06	686/6678/41.671875/8.466836e-06	671/135838/784.437500/8.765321e-06	596/124310/694.171875/9.913071e-06
10	50,000	5472/44114/616.703125/9.804676e-06	892/10433/118.531250/6.674902e-06	1030/255267/3083.531250/8.165186e-06	1065/260264/3294.484375/9.877143e-06
11	1000	0/1/0.000000/0.000000e+00	0/1/0.000000/0.000000e+00	0/1/0.000000/0.000000e+00	0/1/0.000000/0.000000e+00
11	5000	0/1/0.000000/0.000000e+00	0/1/0.000000/0.000000e+00	0/1/0.062500/0.000000e+00	0/1/0.000000/0.000000e+00
11	10,000	0/1/0.000000/0.000000e+00	0/1/0.062500/0.000000e+00	0/1/0.000000/0.000000e+00	0/1/0.000000/0.000000e+00
11	50,000	0/1/0.015625/0.000000e+00	0/1/0.015625/0.000000e+00	0/1/0.031250/0.000000e+00	0/1/0.031250/0.000000e+00

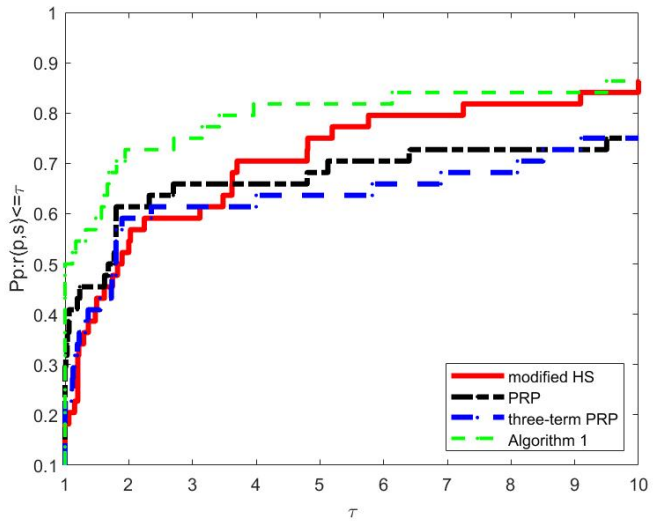

Figure 1. Performance profile based on NI (number of iterations).

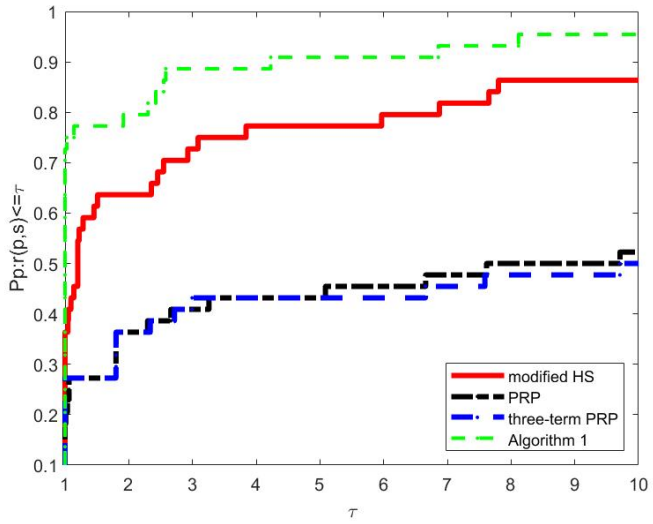

Figure 2. Performance profile based on NF (number of function evaluations).

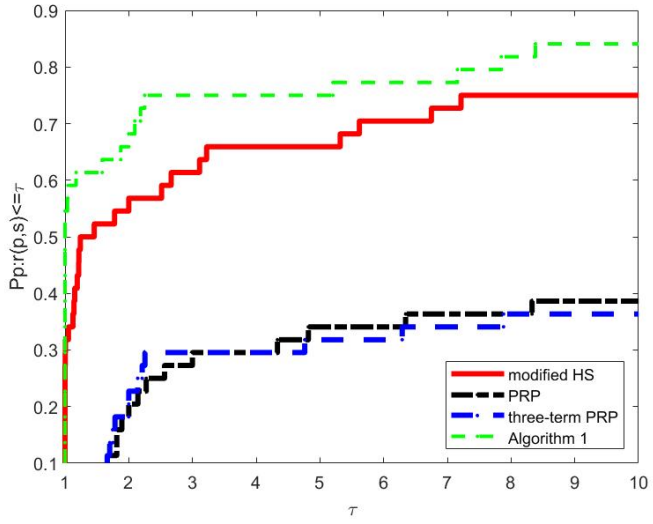

Figure 3. Performance profile based on CPU time (CPU seconds consumed).

Per issue, four dimensions of 1000, 5000, 10,000 and 50,000 were considered. In order to provide a graphical and well-rounded comparison of these treatments, this paper utilizes the performance curves of [34], which have a large resource for messages on both efficiencies and soundness. We graphed testing issue scores P individually, where the measure differs from the corresponding best method by a factor of τ. The performance curves are depicted in Figures 1–3, thereafter. The numerical consequences are tabulated in Table 1, where "Dim" denotes dimension and $\|g_k\|$ denotes the final value.

As can be seen from Figure 1, Algorithm 1 is the most efficient in terms of the sense of the number of iterations, solving 50% of the problem with the least number of iterations. From Figure 2, it can be seen that the most efficient method in the sense of the number of function evaluations is Algorithm 1, which solves 75% of the problem with the minimum number of function evaluations. Figure 3 shows

that Algorithm 1 is also the most efficient algorithm in terms of CPU time considerations.

It can be seen that Algorithm 1 fared splendidly in dealing with large-scale constrained monotone equations. Conclusively, numerical investigations indicate that the suggested algorithm constitutes an effective vehicle for solving systems of convex constrained monotone equations.

4.2. Unconstrained optimization

Numerical outcomes with the proposed Algorithm 2 and modified HS method [29], three-term PRP method [31], and conventional PRP method [21, 32] are reported separately. The fifty problems for the experiment were taken from [35], and the initial points for every question have been given as shown in Table 2. All algorithms use the weak Wolfe-Powell line-search technique. Each test question's termination conditions are

$$\frac{|f_k - f_{k+1}|}{|f_k|} \leq 10^{-5},$$

or

$$\|F_k\| \leq 10^{-6}.$$

The parameters in Algorithm 2 are chosen as: $\zeta = 0.5$, $\xi = 0.95$, $\nu_1 = 2, \nu_2 = 30,000$, and $\nu_3 = 1000$. For each problem, we considered three dimensions of 30,000,12,000 and 300,000. The consumed performance curves are shown in Figures 4–6.

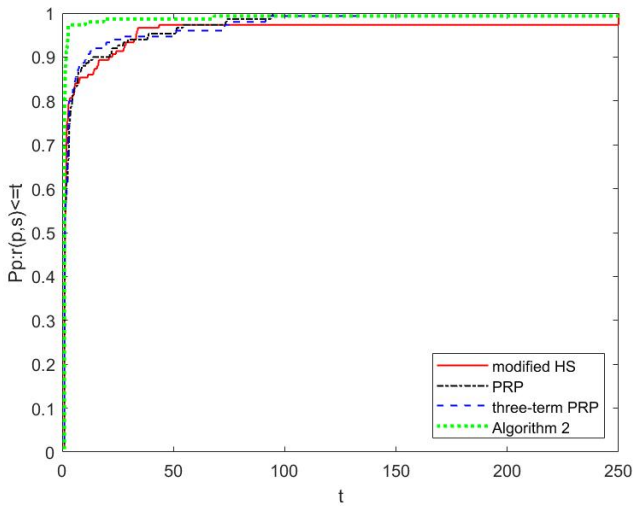

Figure 4. Performance profile based on NI (number of iterations).

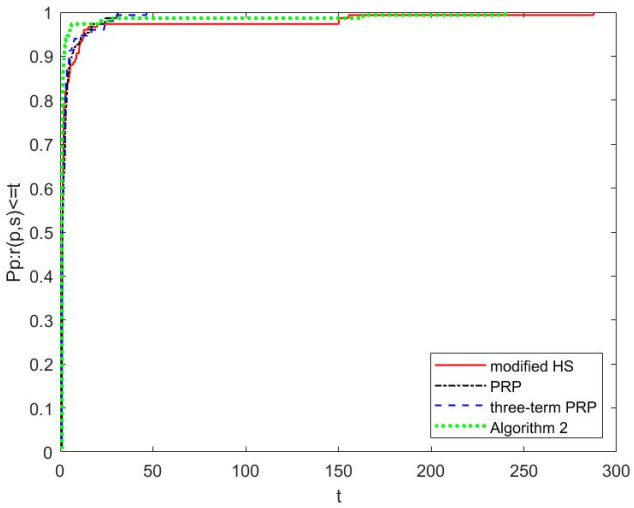

Figure 5. Performance profile based on NFG (the sum of the total iterations and gradient iterations).

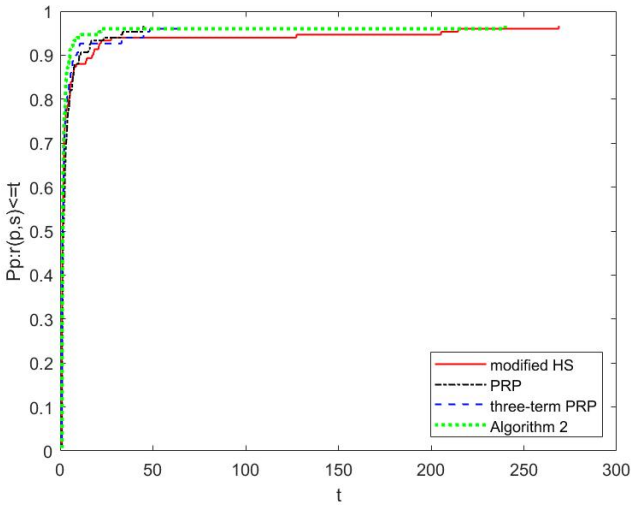

Figure 6. Performance profile based on CPU time (CPU seconds consumed).

Table 2. The test problem.

NO	Problem	x_0	NO	Problem	x_0
1	Extended Freudenstein	$[0.5, -2, ..., 0.5, -2]$	26	Extended Tridiagonal-2 Function	$[1, 1, ..., 1]$
2	Extended Trigonometric Function	$[0.2, 0.2, ..., 0.2]$	27	ARWHEAD (CUTE)	$[1, 1, ..., 1]$
3	Extended Beale Function U63 (MatrixRom)	$[1, 0.8, ..., 1, 0.8]$	28	NONDIA (Shanno-78) (CUTE)	$[-1, -1, ..., -1]$
4	Extended Penalty Function	$[1, 2, 3, ..., n]$	29	EG2 (CUTE)	$[1, 1, ..., 1]$
5	Raydan 1 Function	$[1, 1, ..., 1]$	30	DIXMAANB (CUTE)	$[2, 2, ..., 2]$
6	Raydan 2 Function	$[1, 1, ..., 1]$	31	DIXMAANC (CUTE)	$[2, 2, ..., 2]$
7	Diagonal 1 Function	$[1/n, 1/n, ..., 1/n]$	32	DIXMAANE (CUTE)	$[2, 2, ..., 2]$
8	Diagonal 2 Function	$[1/1, 1/2, ..., 1/n]$	33	Broyden Tridiagonal	$[-1, -1, ..., -1]$
9	Diagonal 3 Function	$[1, 1, ..., 1]$	34	EDENSCH Function (CUTE)	$[0, 0, ..., 0]$
10	Hager Function	$[1, 1, ..., 1]$	35	VARDIM Function (CUTE)	$[1 - 1/n, 1 - 2/n, ..., 1 - n/n]$
11	Generalized Tridiagonal-1 Function	$[2, 2, ..., 2]$	36	LIARWHD (CUTE)	$[4, 4, ..., 4]$
12	Extended Three Exponential Terms	$[0.1, 0.1, ..., 0.1]$	37	DIAGONAL 6	$[1, 1, ..., 1]$
13	Generalized Tridiagonal-2	$[-1, -1, ..., -1, -1]$	38	DIXMAANF (CUTE)	$[2, 2, ..., 2]$
14	Diagonal 4 Function	$[1, 1, ..., 1, 1]$	39	DIXMAANG (CUTE)	$[2, 2, ..., 2]$
15	Diagonal 5 Function (MatrixRom)	$[1.1, 1.1, ..., 1.1]$	40	DIXMAANH (CUTE)	$[2, 2, ..., 2]$
16	Extended Himmelblau Function	$[1, 1, ..., 1]$	41	DIXMAANI (CUTE)	$[2, 2, ..., 2]$
17	Generalized PSC1 Function	$[3, 0.1, ..., 3, 0.1]$	42	DIXMAANJ (CUTE)	$[2, 2, ..., 2]$
18	Extended PSC1 Function	$[3, 0.1, ..., 3, 0.1]$	43	DIXMAANK (CUTE)	$[2, 2, ..., 2]$
19	Extended Block Diagonal BD1 Function	$[0.1, 0.1, ..., 0.1]$	44	DIXMAANL (CUTE)	$[2, 2, ..., 2]$
20	Extended Cliff	$[0, -1, ..., 0, -1]$	45	DIXMAAND (CUTE)	$[2, 2, ..., 2]$
21	Extended Wood Function	$[-3, -1, -3, -1, ...]$	46	ENGVAL1 (CUTE)	$[2, 2, ..., 2]$
22	Extended Quadratic Penalty QP1 Function	$[1, 1, ..., 1]$	47	FLETCHCR (CUTE)	$[0, 0, ..., 0]$
23	Extended Quadratic Penalty QP2 Function	$[1, 1, ..., 1]$	48	COSINE (CUTE)	$[1, 1, ..., 1]$
24	A Quadratic Function QF2	$[0.5, 0.5, ..., 0.5]$	49	Extended DENSCHNB (CUTE)	$[1, 1, ..., 1]$
25	Extended EP1 Function	$[1.5, 1.5, ..., 1.5]$	50	Extended DENSCHNF (CUTE)	$[2, 0, 2, 0, ..., 2, 0]$

4.3. Image restoration

The image restoration problems were solved by using algorithms. It is well known that in bio-engineering, medicine, and other fields of the science and engineering, image restoration techniques play a pivotal role. One of common image degradation models is defined by the following:

$$\min \check{v}(x) = \frac{1}{2}\|\upsilon - \beth x\|_2^2 + \iota\|x\|_1, \tag{4.1}$$

where $\upsilon \in R^{m_1}$ is the observation, Ξ is the linear function of order $m_1 \times m_2$, and ι is a constant greater than zero.

The regularized model (4.1) has attracted much concern, and several scholars have proposed various iterative methods to deal with it [36–38].

Figueiredo [39] developed a gradient-based projection program by reformulating (4.1) as a constrained quadratic program. The reformulation in Figueiredo et al. is by following: split vector x into two pieces, i.e., $x = s - t$, where $s_i = (x_i)_+$, $t_i = (-x_i)_-$, and $(\bullet)_+ = \max\{0, \bullet\}$. Then (4.1) is transformed into

$$\min_{\kappa \geq 0} \frac{1}{2}\kappa^T \Phi \kappa + \gamma^T \kappa, \tag{4.2}$$

$$\kappa = [s, t]^T, \gamma = \iota e_{2m_2} + [-\beth^T \upsilon \; \beth^T \upsilon]^T, \Phi = \begin{bmatrix} \beth^T \beth & -\beth^T \beth \\ -\beth^T \beth & \beth^T \beth \end{bmatrix}.$$

Further, Xiao et al. [40] found (4.2) to be equivocated to the following system on nonlinear equations:

$$H(\kappa) = \min\{\kappa, \Phi\kappa + \gamma\} = 0. \tag{4.3}$$

Pang [41] proved that H satisfies (2.9), while Xiao [40] proved that it satisfies (2.2) too.

Codes' stop condition

$$\frac{|\check{v}(u_{k+1}) - \check{v}(u_k)|}{|\check{v}(u_k)|} \leq 10^{-3},$$

or

$$\frac{\|u_{k+1} - u_k\|}{\|u_k\|} \leq 10^{-3}.$$

Table 3. CPU time results for different algorithms for gray images.

		Algorithm 1	Modified HS	Three-term PRP	NPRP2
Man (1024 × 1024)	0.3	6.6875	24.28125	16.375	33.303140
	0.7	13.09375	32.3125	42.3125	95.854068
Baboon (512 × 512)	0.3	1.6875	6.140625	3.90625	7.276346
	0.7	2.765625	9.28125	12.14062	22.251381

In the experiments, "Man"," Baboon", "colorcheckertestimage", and "car" are the tested images. Comparing Algorithm 1, modified HS [29], three-term PRP [31], and NPRP2 [42], the four algorithms are not effective at adding 30%, 70% to the noise. The parameters in Algorithm 1 are chosen as: $\varpi = 0.01$, $\vartheta = 0.8$, $\tilde{q} = 1$, $\dot{t} = 0.5$, $\nu_1 = 2.2, \nu_2 = 3000$, and $\nu_3 = 1200$. The time taken for each

algorithm is summarized in Tables 3 and 4. From Tables 3 and 4, it can be seen that CPU time for both contrastive and color images processed with Algorithm 1 is less than the time used by the other algorithms, which shows that the algorithms proposed in this paper are more advantageous in dealing with grey image and color image restoration problems.

Table 4. CPU time results for different algorithms for color images.

		Algorithm 1	Modified HS	Three-term PRP	NPRP2
colorcheckertestimage (1541 × 1024)	0.3	9.077376	12.136426	17.377901	16.661125
	0.7	14.084278	37.827917	37.333502	30.093593
car (3504 × 2336)	0.3	41.74019	62.762366	96.119989	91.296092
	0.7	55.1054	160.855	219.161032	163.100240

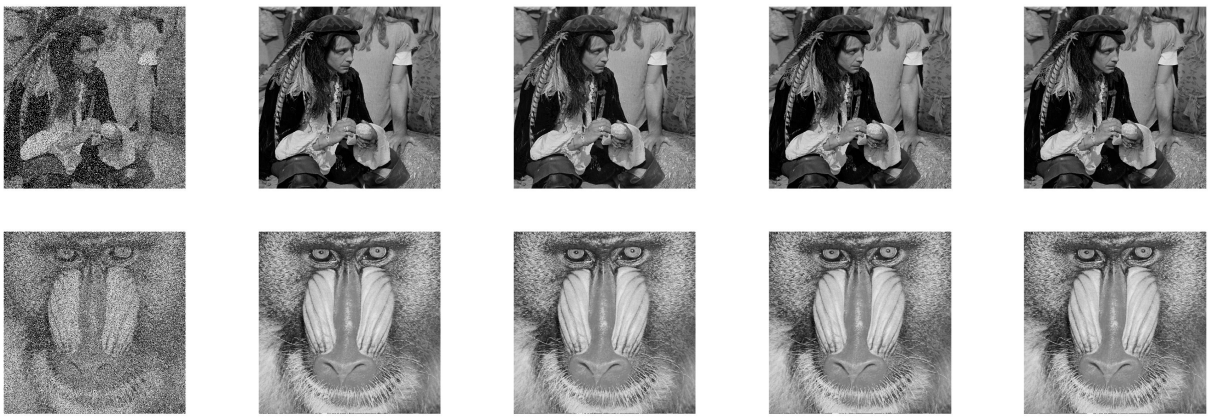

Figure 7. From front to back for each row: the images with 30% image noise added, the images processed by Algorithm 1, the modified HS method, the three-term PRP method, and the NPRP2 method.

Figure 8. From front to back for each row: the images with 70% image noise added, the images processed by Algorithm 1, the modified HS method, the three-term PRP method, and the NPRP2 method.

Figures 7 and 8 show that Algorithm 1 has performed well in tackling the gray image restoration.

Visualizing the outcomes of color, Figures 9 and 10 are given and from the results, it is clear that the algorithm of this paper is more competitive in dealing with color image restoration. In image restoration experiments, we usually use the PSNR value and SSIM value to estimate the quality of the processed image. The higher the PSNR value, the less distorted the image is. The higher the SSIM value, the better the image setup. When two mirrors are identical, the SSIM value is 1. As can be seen from Figures 9 and 10, the SSIM value obtained by Algorithm 1 is relatively small compared to the traditional algorithm, and the image restoration effect is not as good as the traditional algorithm. This aspect will be investigated in the future.

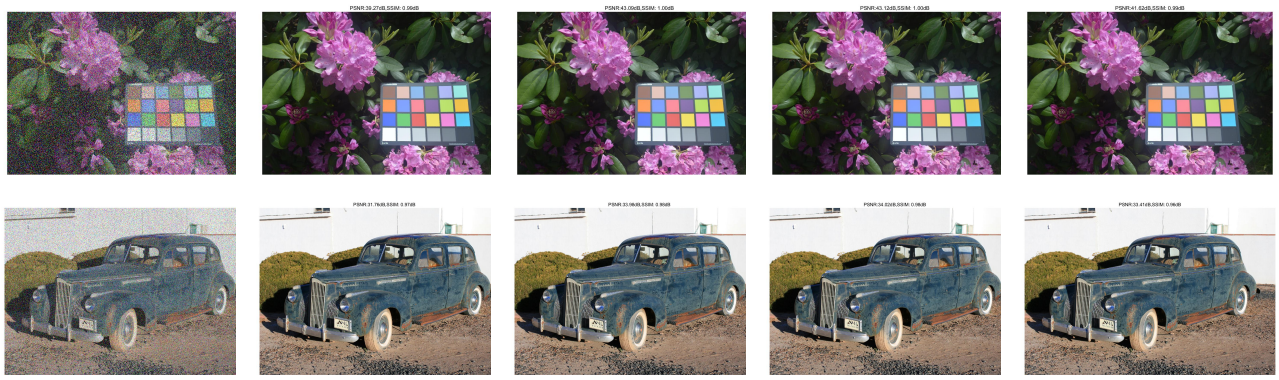

Figure 9. From front to back for each row: the images with 30% image noise added, the images processed by Algorithm 1, the modified HS method, the three-term PRP method, and the NPRP2 method.

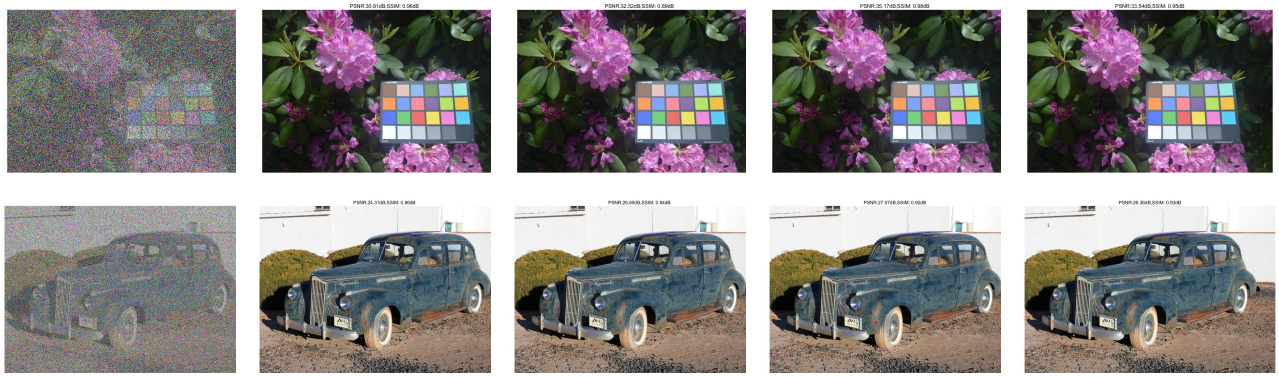

Figure 10. From front to back for each row: the images with 70% image noise added, the images processed by Algorithm 1, the modified HS method, the three-term PRP method, and the NPRP2 method.

4.4. Machine learning

We now embed the improved HS method into a stochastic large subspace algorithm [43] by replacing its search direction with (2.8), thus building an improved HS algorithm based on the stochastic large subspace and variance-reducing SCGN-based algorithms (called mSCGN and mSDSCGN). Next, we test and compare the stochastic gradient descent algorithm (SGD) [44] and the STOCHASTIC

VARIANCE GRADIENT REDUCTION (SVRG) [45] on the following two learning models:

(i) Nonconvex SVM model with a sigmoid loss function:

$$\min_{x\in\Re^d} \frac{1}{t}\sum_{s=0}^{t} f_s(x) + \lambda\|x\|^2 \tag{4.4}$$

where $f_s(x) = 1 - tanh(\omega_s < x, \varpi_s >)$, $\varpi \in \Re$, $\omega \in \{-1, 1\}$ signify feature vectors and the corresponding labels, respectively. Support vector machine models were implemented in different areas for pattern recognition, currently focusing on information categorization, encompassing text and images.

(ii) Nonconvex regularized ERM model with a nonconvex sigmoid loss function:

$$\min_{x\in\Re^d} \frac{1}{t}\sum_{s=0}^{t} f_s(x) + \frac{\lambda}{2}\|x\|^2 \tag{4.5}$$

where $f_s(x) = 1/\{1 + exp(\epsilon_s\varepsilon_s^T x)\}$, $\epsilon_s \in \{-1, 1\}$ is target value of the s-th sample, $\varepsilon_s \in \Re^d$ is the eigenvector of the s-th sample, and $\lambda > 0$ is the regularization parameter. Binary classification models are instrumental in practical problems. While a non-convex ERM model concentrates on minimizing classification inaccuracy, the s-type function is also exhibited to be usually superior to alternative loss functions. Accordingly, nonconvex ERM models with s-type loss functions are valuable for mathematical modeling.

The parameters in mSCGN and mSDSCGN are chosen as: $\nu_1 = 3.8, \nu_2 = 300$, and $\nu_3 = 100$. The dataset comes from https://www.csie.ntu.edu.tw cjlin/libsvmtools/datasets/. All algorithms were executed on three large datasets, with the specific details of the datasets are shown in Table 5. In all experiments, the regularization was chosen to be $\lambda = 10^{-5}$ and a similar batch size m was used in each iteration. Gradient $\nabla f(x)$ and the Hessian matrix $\nabla^2 f(x)$ are estimated by:

$$\nabla f(x)_s = \frac{f(x + \sigma e_s) - f(x - \sigma e_s)}{2\sigma},$$

$$\nabla^2 f(x)_{s,r} = \frac{f(x + \sigma e_s + \sigma e_r) - f(x + \sigma e_s) - f(x + \sigma e_r) + f(x)}{\sigma^2},$$

where $\sigma = 10^{-4}$ and e_s is the s-th unit-vector. Let the regularization factor λ of the learning model be 10^{-4}, and the convergence result of the algorithms is shown in Figures 11 and 12. In this experiment, the largest count for internal iterations is 30. In principle, the lower the value of the curve, the better convergence in the corresponding algorithm.

Table 5. Details of data sets.

Data set	Training samples (t)	Dimensions (d)
Adult	32,562	123
IJCNN	49,990	22
Covtype	581,012	54

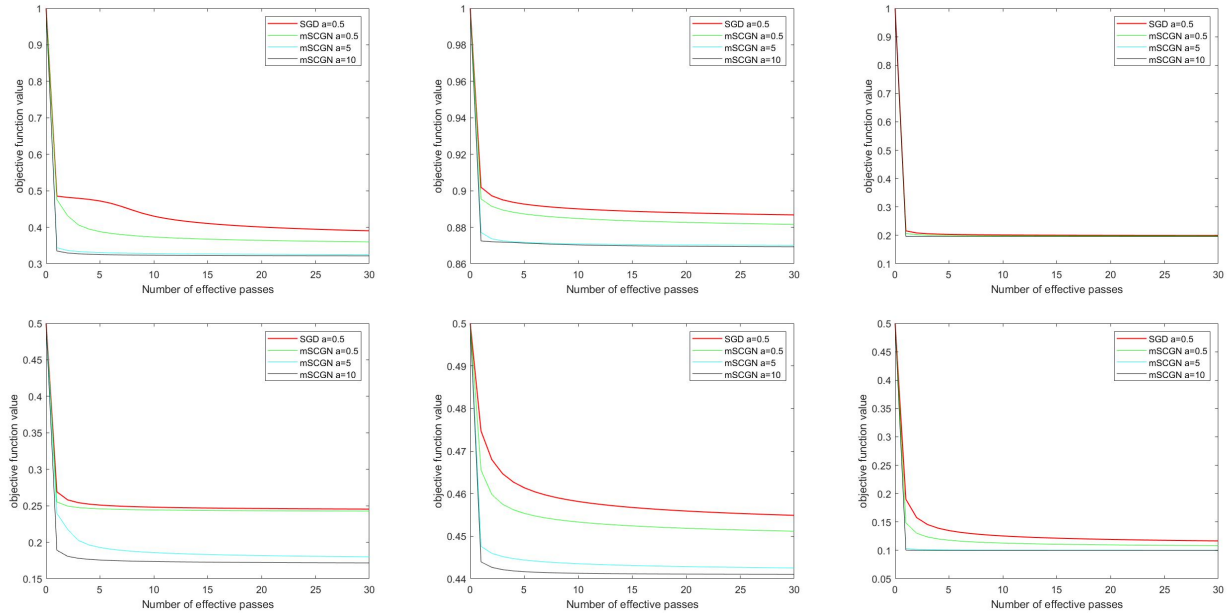

Figure 11. Numerical performance of SGD and mSCGN for solving models 1 and 2 in three datasets.

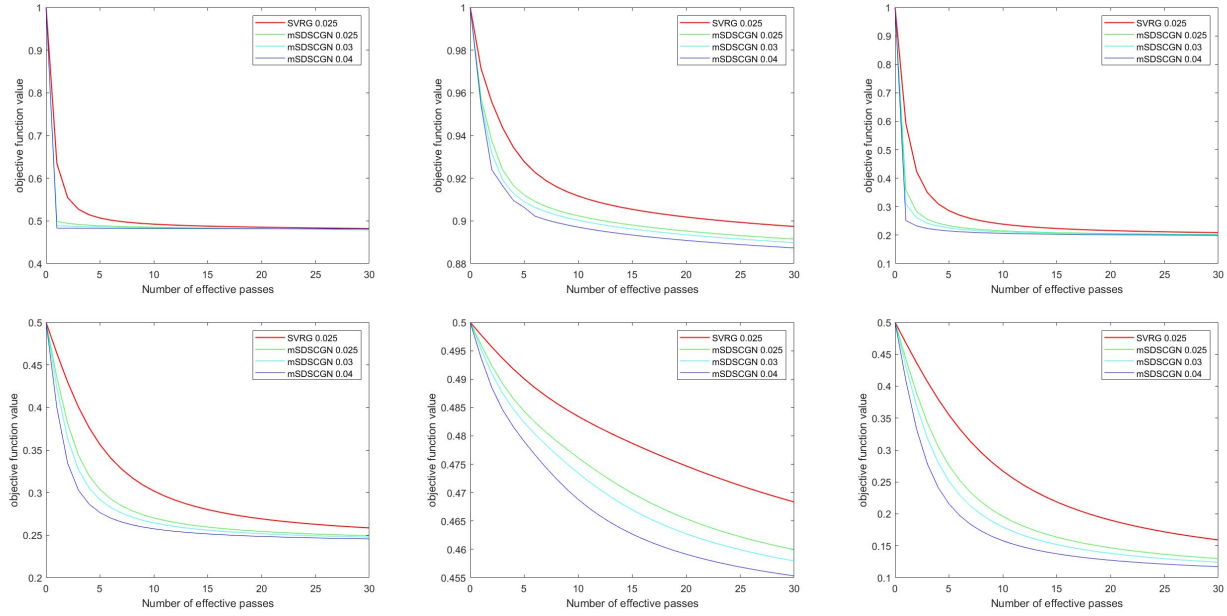

Figure 12. Numerical performance of SVRG and mSCGN for solving models 1 and 2 in three datasets.

The different algorithms' behavior is revealed by plotting the curve of function values over the iteration. Figure 11 illustrates the behavior of mSCGN and SGD at solving (4.4) and (4.5) with alternative decreasing steps. As verified from Figure 11, both algorithms managed to tackle the models. The function values decline at a greater rate initially and incrementally remain constant. Notice that when the mSCGN algorithm and the SGD algorithm share the same step-size on the datasets, the former

decays the function value firmer. This reveals that the algorithm herein designed is well executed for having sufficient descent and trust domain capabilities. Observe that for mSCGN, $\alpha_k = 10k^{-0.9}$ converges quicker. Our algorithm exhibits superiority at appropriate step sizes, and experimental results perform better at larger decreasing step sizes.

Figure 12 illustrates the behavior of mSDSCGN and SVRG for solving (4.4) and (4.5) at separate constant step sizes. Analyzing Figure 12, it is obvious that even though the SVRG algorithm uses the best step size, SVRG does not perform as well as mSDSCGN on all three datasets. We observed that of those step sizes, 0.04 is optimally suited toward mSDSCGN. Overall, our algorithm shows more promise and efficiency than others. Based on the analysis of the results, one can conclude that our suggested method is useful for machine learning.

5. Conclusions

We propose a modified HS conjugate gradient method with the following characteristics: 1) Fulfills descent and trust-region features. 2) A particular method's global convergence for the non-convex functions can be easily established. 3) The image restoration, machine learning issues, and nonlinear monotone equations experiments were tested and the outcomes indicated that the algorithms have promising numerical characterizations for tackling these issues.

Use of AI tools declaration

The authors declare they have not used Artificial Intelligence (AI) tools in the creation of this article.

Acknowledgement

This work is supported by the Guangxi Science and Technology Base and Talent Project (Grant No. AD22080047), the National Natural Science Foundation of Guangxi Province (Grant No. 2023GXN-FSBA026063), the major talent project of Guangxi (GXR-6BG242404), the Bagui Scholars Program of Guangxi, and the 2024 Graduate Innovative Programs Establishment (Grant No. ZX01030031124006). We are sincerely grateful to the editors and reviewers for their valuable comments, which have further enhanced this paper.

Conflict of interest

The authors declare there are no conflicts of interest.

References

1. J. M. Ortega, W. C. Rheinboldt, *Iterative Solution of Nonlinear Equation in Sseveral Variables*, Society for Industrial and Applied Mathematics, Philadelpia, PA, USA, **598** (1970).
2. W. C. Rheinboldt, *Methods for Solving Systems of Nonlinear Equations*, Society for Industrial and Applied Mathematics, 1998.

3. L. Qi, J. Sun, A nonsmooth version of Newton's method, *Math. Program.*, **58** (1993), 353–367. https://doi.org/10.1007/BF01581275
4. J. E. Dennis, J. J. Moré, A characterization of superlinear convergence and its application to quasi-Newton methods, *Math. Comput.*, **28** (1974), 549–560. https://doi.org/10.1090/S0025-5718-1974-0343581-1
5. J. E. Jr. Dennis, J. J. Moré, quasi-Newton methods, motivation and theory, *SIAM Rev.*, **19** (1977), 46–89. https://doi.org/10.1137/1019005
6. X. Tong, L. Qi, Y. Yang, The Lagrangian globalization method for nonsmooth constrained equations, *Comput. Optim. Appl.*, **33** (2006), 89–109. https://doi.org/10.1007/s10589-005-5960-9
7. W. Zhou, D. Li, A globally convergent BFGS method for nonlinear monotone equations without any merit functions, *Math. Comput.*, **77** (2008), 2231–2240. https://doi.org/10.1090/S0025-5718-08-02121-2
8. C. Wang, Y. Wang, A superlinearly convergent projection method for constrained systems of nonlinear equations, *J. Global Optim.*, **44** (2009), 283–296. https://doi.org/10.1007/s10898-008-9324-8
9. A. B. Abubakar, P. Kumam, An improved three-term derivative-free method for solving nonlinear equations, *Comput. Appl. Math.*, **37** (2018), 6760–6773. https://doi.org/10.1007/s40314-018-0712-5
10. A. M. Awwal, P. Kumam, A. B. Abubakar, A modified conjugate gradient method for monotone nonlinear equations with convex constraints, *Appl. Numer. Math.*, **145** (2019), 507–520. https://doi.org/10.1016/j.apnum.2019.05.012
11. A. H. Ibrahim, A. I. Garba, H. Usman, J. Abubakar, A. B. Abubakar, Derivative-free RMIL conjugate gradient method for convex constrained equations, *Thai J. Math.*, **18** (2019), 212–232.
12. J. Liu, Y. Feng, A derivative-free iterative method for nonlinear monotone equations with convex constraints, *Numerical Algorithms*, **82** (2019), 245–262. https://doi.org/10.1007/s11075-018-0603-2
13. Y. Xiao, H. Zhu, A conjugate gradient method to solve convex constrained monotone equations with applications in compressive sensing, *J. Math. Anal. Appl.*, **405** (2013), 310–319. https://doi.org/10.1016/j.jmaa.2013.04.017
14. M. V. Solodov, B. F. Svaiter, A globally convergent inexact Newton method for systems of monotone equations, in *Reformulation: Nonsmooth, Piecewise Smooth, Semismooth and Smoothing Methods. Applied Optimization*, Springer, Boston, MA, **22** (1999), 355–369. https://doi.org/10.1007/978-1-4757-6388-1_18
15. G. Yuan, P. Li, J. Lu, The global convergence of the BFGS method with a modified WWP line search for nonconvex functions, *Numerical Algorithms*, **91** (2022), 353–365. https://doi.org/10.1007/s11075-022-01265-3
16. G. Yuan, M. Zhang, Y. Zhou, Adaptive scaling damped BFGS method without gradient Lipschitz continuity, *Appl. Math. Lett.*, **124** (2022), 107634. https://doi.org/10.1016/j.aml.2021.107634
17. M. V. Solodov, B. F. Svaiter, A new projection method for variational inequality problems, *SIAM J. Control Optim.*, **37** (1999), 765–776. https://doi.org/10.1137/S0363012997317475
18. Y. Dai, Y. Yuan, A nonlinear conjugate gradient method with a strong global convergence property, *SIAM J. Control Optim.*, **10** (1999), 177–182. https://doi.org/10.1137/S1052623497318992

19. 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
20. M. R. Hestenes, E. Stiefel, Methods of conjugate gradients for solving linear systems, *J. Res. Natl. Bur. Stand.*, **49** (1952), 409–436. https://doi.org/10.6028/JRES.049.044
21. S. Rouge, E. Polak, G. Ribière, Note sur la convergence de méthodes de directions conjuguées, *Rev. Fr. Inf. Rech. Opérationnelle. Série Rouge*, **3** (1969), 35–43. https://doi.org/10.1051/M2AN/196903R100351
22. L. Zhang, W. Zhou, D. Li, Some descent three-term conjugate gradient methods and their global convergence, *Optim. Methods Software*, **22** (2007), 697–711. https://doi.org/10.1080/10556780701223293
23. G. Yuan, H. Yang, M. Zhang, Adaptive three-term PRP algorithms without gradient Lipschitz continuity condition for nonconvex functions, *Numerical Algorithms*, **91** (2022), 145–160. https://doi.org/10.1007/s11075-022-01257-3
24. X. Wu, H. Shao, P. Liu, Y. Zhuo, An inertial spectral CG projection method based on the memoryless BFGS update, *J. Optim. Theory Appl.*, **198** (2023), 1130–1155. https://doi.org/10.1007/s10957-023-02265-6
25. 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. Modell.*, **118** (2023), 393–411. https://doi.org/10.1016/j.apm.2023.01.018
26. R. Huang, Y. Qin, K. Liu, G. Yuan, Biased stochastic conjugate gradient algorithm with adaptive step size for nonconvex problems, *Expert Syst. Appl.*, **238** (2024), 121556. https://doi.org/10.1016/j.eswa.2023.121556
27. C. Ouyang, C. Lu, X. Zhao, R. Huang, G. Yuan, Y. Jiang, Stochastic three-term conjugate gradient method with variance technique for non-convex learning, *Stat. Comput.*, **34** (2024), 107. https://doi.org/10.1007/s11222-024-10409-5
28. G. Yuan, X. Wang, Z. Sheng, Family weak conjugate gradient algorithms and their convergence analysis for nonconvex functions, *Numerical Algorithms*, **84** (2020), 935–956. https://doi.org/10.1007/s11075-019-00787-7
29. W. Hu, J. Wu, G. Yuan, Some modified Hestenes-Stiefel conjugate gradient algorithms with application in image restoration, *Appl. Numer. Math.*, **158** (2020), 360–376. https://doi.org/10.1016/j.apnum.2020.08.009
30. Q. Li, D. Li, A class of derivative-free methods for large-scale nonlinear monotone equations, *IMA J. Numer. Anal.*, **31** (2011), 1625–1635. https://doi.org/10.1093/imanum/drq015
31. G. Wu, Y. Li, G. Yuan, A three-term conjugate gradient algorithm with quadratic convergence for unconstrained optimization problems, *Math. Probl. Eng.*, **2018** (2018). https://doi.org/10.1155/2018/4813030
32. B. T. Polyak, The conjugate gradient method in extremal problems, *USSR Comput. Math. Math. Phys.*, **9** (1969), 94–112. https://doi.org/10.1016/0041-5553(69)90035-4
33. G. Yuan, Z. Wei, S. Lu, Limited memory BFGS method with backtracking for symmetric nonlinear equations, *Math. Comput. Modell.*, **54** (2011), 367–377. https://doi.org/10.1016/j.mcm.2011.02.021

34. E. D. Dolan, J. J. Moré, Benchmarking optimization software with performance profiles, *Math. Program.*, **91** (2002), 201–213. https://doi.org/10.1007/s101070100263
35. N. Andrei, An unconstrained optimization test functions collection, *Adv. Model. Optim.*, **10** (2008), 147–161.
36. M. A. T. Figueiredo, R. D. Nowak, An EM algorithm for wavelet-based image restoration, *IEEE Trans. Image Process.*, **12** (2003), 906–916. https://doi.org/10.1109/TIP.2003.814255
37. C. De Mol, M. Defrise, A note on wavelet-based inversion algorithms, *Contemp. Math.*, **313** (2002), 85–96. https://doi.org/10.1090/conm/313/05370
38. J. Yang, W. Yin, Y. Zhang, Y. Wang, A fast algorithm for edge-preserving variational multichannel image restoration, *SIAM J. Imag. Sci.*, **2** (2009), 569–592. https://doi.org/10.1137/080730421
39. M. A. T. Figueiredo, R. D. Nowak, S. J. Wright, Gradient projection for sparse reconstruction: Application to compressed sensing and other inverse problems, *IEEE J. Sel. Top. Signal Process.*, **1** (2007), 586–597. https://doi.org/10.1109/JSTSP.2007.910281
40. Y. Xiao, Q. Wang, Q. Hu, Non-smooth equations based method for ℓ_1-norm problems with applications to compressed sensing, *Nonlinear Anal. Theory Methods Appl.*, **74** (2011), 3570–3577. https://doi.org/10.1016/j.na.2011.02.040
41. J. Pang, Inexact Newton methods for the nonlinear complementarity problem, *Math. Program.*, **36** (1986), 54–71. https://doi.org/10.1007/BF02591989
42. A. Mousavi, M. Esmaeilpour, A. Sheikhahmadi, A new family of Polak-Ribière-Polyak conjugate gradient method for impulse noise removal, *Soft Comput.*, **27** (2023), 17515–17524. https://doi.org/10.1007/s00500-023-09232-3
43. G. Yuan, Y. Zhou, L. Wang, Q. Yang, Stochastic bigger subspace algorithms for nonconvex stochastic optimization, *IEEE Access*, **9** (2021), 119818–119829. https://doi.org/10.1109/ACCESS.2021.3108418
44. L. Bottou, Large-scale machine learning with stochastic gradient descent, in *Proceedings of COMPSTAT'2010: 19th International Conference on Computational StatisticsParis France, August 22-27, 2010 Keynote, Invited and Contributed Papers*, (2010), 177–186. https://doi.org/10.1007/978-3-7908-2604-3_16
45. R. Johnson, T. Zhang, Accelerating stochastic gradient descent using predictive variance reduction, *Adv. Neural Inf. Process. Syst.*, **26** (2013).

AIMS Press