



*Research article***An improved LS-RMIL-type conjugate gradient projection algorithm for systems of nonlinear equations and impulse noise image restoration****Yan Xia, Xuejie Ma, and Dandan Li***

School of Artificial Intelligence, Guangzhou Huashang College, Guangzhou 511300, China

* **Correspondence:** Email: liddjq@gdhsc.edu.cn.

Abstract: This paper proposes an improved LS-RMIL-type conjugate gradient projection algorithm designed for solving systems of nonlinear equations with convex constraints. The algorithm introduces a search direction that maintains sufficient descent and trust-region properties independent of the line search approach. It operates under relatively mild conditions, requiring only continuity and monotonicity of nonlinear equations, thus avoiding the need for stronger assumptions such as Lipschitz continuity. The global convergence of the algorithm is established under these relaxed conditions. Furthermore, numerical experiments demonstrate that the algorithm exhibits superior efficiency and stability, particularly in solving large-scale nonlinear systems and in applications such as impulse noise image restoration, outperforming existing methods.

Keywords: conjugate gradient projection method; systems of nonlinear equations; global convergence; impulse noise image restoration

Mathematics Subject Classification: 65K05, 90C56

1. Introduction

Systems of nonlinear equations are fundamental to a diverse range of applications, including power flow analysis [1], economic equilibrium modeling [2], the development of generalized Bregman distance proximal point methods [3], and traffic assignment [4]. Meanwhile, these systems also involve monotone variational inequalities [5, 6] and compression sensing problems [7, 8]. Given the ubiquity and significance of such problems across these varied domains, the study and development of numerical methods to efficiently solve systems of nonlinear equations are of considerable practical importance. In this paper, we focus on a specific class of systems of nonlinear equations subject to convex constraints, which can be formulated as follows:

$$\theta(a) = 0, \quad a \in \Theta, \quad (1.1)$$

where $\Theta \subseteq \mathbb{R}^n$ is a non-empty, and closed convex set. The function $\theta : \mathbb{R}^n \rightarrow \mathbb{R}^n$ is assumed to possess monotonicity and continuous differentiability, satisfying the following condition:

$$\langle \theta(a) - \theta(b), a - b \rangle \geq 0, \quad \forall a, b \in \mathbb{R}^n.$$

Generally, the gradient-type method generates a sequence $\{a_k\}$, defined as follows:

$$a_{k+1} = a_k + t_k d_k, \quad k \geq 0,$$

where t_k is the step length, and d_k denotes the search direction. The choice of the search direction d_k gives rise to various gradient-type methods, such as the steepest descent methods, Newton's methods, and quasi-Newton methods [9, 10]. Newton's and quasi-Newton methods, along with their numerous variants, have been extensively studied due to their strong local linear convergence properties. For instance, Mahdavi et al. [11] proposed and analyzed a nonmonotone quasi-Newton algorithm for strongly convex multiobjective optimization, demonstrating its global convergence and local superlinear convergence rate under certain conditions. Sihwail et al. [12] proposed a novel hybrid method, Newton-Harris hawks optimization, which combines Newton's methods and Harris hawks optimization to effectively solve systems of nonlinear equations. Moreover, Krutikov et al. [13] demonstrated that quasi-Newton methods, when applied to strongly convex functions with a Lipschitz gradient, achieve geometric convergence without relying on local quadratic approximations. However, despite these advantages, Newton's and quasi-Newton methods involve the computation of the Hessian matrix or its approximation value at each iteration, which significantly increases computational complexity. This requirement can be a limiting factor, particularly for large-scale problems where the Hessian matrix is difficult to compute and store efficiently.

The conjugate gradient method [14, 15] is one of the most effective approaches in the field of the gradient-type methods. It is highly recognized for its efficiency, simplicity, lower storage requirements, and reliable convergence properties. These characteristics make it particularly well-suited for solving large-scale systems of nonlinear equations [16, 17]. The method's search direction is typically defined as follows:

$$d_k = \begin{cases} -\theta_k, & k = 0, \\ -\theta_k + \beta_k d_{k-1}, & k \geq 1, \end{cases}$$

where $\theta_k \triangleq \theta(a_k)$ and β_k is known as the conjugate parameter. The choice of β_k differentiates various conjugate gradient methods. Several advancements have been made in the development of conjugate gradient methods with different conjugate parameters [18]. For instance, Ma et al. [19] proposed a modified inertial three-term conjugate gradient method for solving nonlinear monotone equations with convex constraints. This method is notable for its global and Q-linear convergence properties, and has demonstrated superior numerical performance in applications such as sparse signal recovery and image restoration in compressed sensing. Furthermore, Liu et al. [20] introduced a spectral conjugate gradient method with an inertial factor for solving nonlinear pseudo-monotone equations over a convex set. Additionally, Sabiu et al. [21] developed an optimal scaled Perry conjugate gradient method for solving large-scale systems of monotone nonlinear equations. This method ensures global convergence under the conditions of monotonicity and Lipschitz continuity.

Inspired by the classical Liu-Story (LS) [22] and Rivai-Mohamad-Ismail-Leong (RMIL) [23] conjugate parameters, as well as incorporating the hybrid technique (e.g., [24, 25]) and the projection

approach, we develop a modified LS-RMIL-type conjugate gradient projection algorithm. The proposed algorithm is specifically designed for solving systems of nonlinear equations with convex constraints. In this paper, $\|\cdot\|$ and $\langle\cdot,\cdot\rangle$ represent the Euclidean norm and the inner product of vectors, respectively.

2. Algorithm design

2.1. Search direction and its properties

In this section, we refine and enhance the search direction employed in the optimization processes of the LS and RMIL methods. Specifically, Liu et al. [22] and Rivai et al. [23] introduced conjugate parameters, defined respectively as:

$$\beta_k^{LS} = \frac{\langle\theta_k, y_{k-1}\rangle}{-\langle\theta_{k-1}, d_{k-1}\rangle}, \quad \beta_k^{RMIL} = \frac{\langle\theta_k, y_{k-1}\rangle}{\|d_{k-1}\|^2}.$$

Based on the insights derived from these parameters, we adopt the hybrid technique (e.g., [24, 25]) that combines their key features. This leads to the formulation of a new conjugate parameter, which is subsequently incorporated into the framework of a three-term search direction. Our primary objective is to construct a novel search direction that ensures both the sufficient descent and trust-region properties, which are critical for the robustness and efficiency of the optimization process. To accomplish this, we carefully design a novel search direction tailored to meet these requirements. The designed search direction d_k is defined as follows:

$$d_k = \begin{cases} -\theta_k, & k = 0, \\ -\theta_k + \beta_k d_{k-1} + \varpi_k y_{k-1}, & k \geq 1, \end{cases} \quad (2.1)$$

where the conjugate parameter β_k and the scalar parameter ϖ_k are given by

$$\beta_k = \frac{\langle\theta_k, y_{k-1}\rangle}{c_k} - \frac{\|y_{k-1}\|^2 \langle\theta_k, d_{k-1}\rangle}{c_k^2} \quad \text{and} \quad \varpi_k = \frac{\nu_k \langle\theta_k, d_{k-1}\rangle}{c_k}, \quad (2.2)$$

with $y_{k-1} = \theta_k - \theta_{k-1}$. One scalar parameter c_k is crucial for maintaining stability in the iterative process, and is defined by

$$c_k = \max \left\{ \mu \|d_{k-1}\| \|y_{k-1}\|, -\langle\theta_{k-1}, d_{k-1}\rangle, \|d_{k-1}\|^2 \right\},$$

where μ is a positive constant. Another scalar parameter ν_k is introduced to fine-tune the adjustment of the search direction. It is defined as $\nu_k = \min\{\tilde{\nu}, \max\{\bar{\nu}_k, 0\}\}$ with $0 < \tilde{\nu} < 1$, and $\bar{\nu}_k = \frac{\langle\theta_k, y_{k-1} - s_{k-1}\rangle}{\|\theta_k\|^2}$, where $s_{k-1} = a_k - a_{k-1}$ represents the difference between the iterative points a_k and a_{k-1} of the optimization variable.

Before delving into the sufficient descent and trust-region properties of the designed search direction (2.1), we can deduce some important bounds from the definition of β_k and ϖ_k . We consider the bound for β_k :

$$|\beta_k| \leq \frac{\|\theta_k\| \|y_{k-1}\|}{c_k} + \frac{\|y_{k-1}\|^2 \|\theta_k\| \|d_{k-1}\|}{c_k^2},$$

which can be further bounded by

$$|\beta_k| \leq \frac{\|\theta_k\| \|y_{k-1}\|}{\mu \|d_{k-1}\| \|y_{k-1}\|} + \frac{\|y_{k-1}\|^2 \|\theta_k\| \|d_{k-1}\|}{\mu^2 \|d_{k-1}\|^2 \|y_{k-1}\|^2}.$$

Simplifying this, we obtain

$$|\beta_k| \leq \left(\frac{1}{\mu} + \frac{1}{\mu^2} \right) \frac{\|\theta_k\|}{\|d_{k-1}\|}. \quad (2.3)$$

Next, we consider the bound for ϖ_k :

$$|\varpi_k| \leq \frac{\nu_k \|\theta_k\| \|d_{k-1}\|}{c_k} \leq \frac{\tilde{\nu}_k \|\theta_k\| \|d_{k-1}\|}{\mu \|d_{k-1}\| \|y_{k-1}\|} = \frac{\tilde{\nu}}{\mu} \frac{\|\theta_k\|}{\|y_{k-1}\|}. \quad (2.4)$$

Lemma 1. *The search direction d_k generated by (2.1) satisfies the sufficient descent property:*

$$\langle \theta_k, d_k \rangle \leq -M \|\theta_k\|^2,$$

where $M = 1 - \frac{1}{4}(1 + \tilde{\nu})^2$.

Proof. For $k = 0$, the conclusion is straightforward, which implies $\langle \theta_0, d_0 \rangle = -\|\theta_0\|^2 \leq -M \|\theta_0\|^2$. For $k \geq 1$, together with the search direction generated by (2.1), we have:

$$\begin{aligned} \langle \theta_k, d_k \rangle &= \langle \theta_k, -\theta_k + \beta_k d_{k-1} + \varpi_k y_{k-1} \rangle \\ &= -\|\theta_k\|^2 + \frac{\langle \theta_k, y_{k-1} \rangle \langle \theta_k, d_{k-1} \rangle}{c_k} - \frac{\|y_{k-1}\|^2 \langle \theta_k, d_{k-1} \rangle^2}{c_k^2} + \frac{\nu_k \langle \theta_k, y_{k-1} \rangle \langle \theta_k, d_{k-1} \rangle}{c_k} \\ &= -\|\theta_k\|^2 + (1 + \nu_k) \frac{\langle \theta_k, y_{k-1} \rangle \langle \theta_k, d_{k-1} \rangle}{c_k} - \frac{\|y_{k-1}\|^2 \langle \theta_k, d_{k-1} \rangle^2}{c_k^2}. \end{aligned} \quad (2.5)$$

By applying the inequality $\langle e_k, g_k \rangle \leq \frac{1}{2}(\|e_k\|^2 + \|g_k\|^2)$, where $e_k = (1 + \nu_k)\theta_k / \sqrt{2}$ and $g_k = \sqrt{2}\langle \theta_k, d_{k-1} \rangle y_{k-1} / c_k$, we obtain the following result:

$$\frac{(1 + \nu_k) \langle \theta_k, y_{k-1} \rangle \langle \theta_k, d_{k-1} \rangle}{c_k} \leq \frac{1}{4}(1 + \nu_k)^2 \|\theta_k\|^2 + \frac{\langle \theta_k, d_{k-1} \rangle^2 \|y_{k-1}\|^2}{c_k^2}. \quad (2.6)$$

Substituting (2.6) into (2.5), we obtain

$$\langle \theta_k, d_k \rangle \leq -\|\theta_k\|^2 + \frac{1}{4}(1 + \tilde{\nu}_k)^2 \|\theta_k\|^2 \leq -\left(1 - \frac{1}{4}(1 + \tilde{\nu})^2\right) \|\theta_k\|^2.$$

Thus, the result holds. \square

Lemma 2. *The search direction d_k generated by (2.1) satisfies the trust-region property:*

$$M \|\theta_k\| \leq \|d_k\| \leq N \|\theta_k\|,$$

where $N = 1 + \frac{1}{\mu} + \frac{1}{\mu^2} + \frac{\tilde{\nu}}{\mu}$.

Proof. From Lemma 1, we have $-\|\theta_k\| \|d_k\| \leq \langle \theta_k, d_k \rangle \leq -M \|\theta_k\|^2$, which implies:

$$\|d_k\| \geq M \|\theta_k\|.$$

Additionally, together with (2.1), we obtain:

$$\|d_k\| = \|-\theta_k + \beta_k d_{k-1} + \varpi_k y_{k-1}\| \leq \|\theta_k\| + |\beta_k| \|d_{k-1}\| + |\varpi_k| \|y_{k-1}\|.$$

Substituting the inequalities $|\beta_k|$ and $|\varpi_k|$ defined in (2.3) and (2.4), respectively, into the above equality, we have

$$\|d_k\| \leq \left(1 + \frac{1}{\mu} + \frac{1}{\mu^2} + \frac{\tilde{\nu}}{\mu}\right) \|\theta_k\|.$$

Thus, the result holds. \square

2.2. Algorithm description

Before delving into the specifics of our proposed algorithm, it is essential to first clarify the line search approach, the projection operator, and the iterative update rule employed in the proposed algorithm. These foundational components play a crucial role in the overall efficacy of the proposed algorithm.

First, in the proposed algorithm, the line search approach is used to determine an appropriate step length $t_k = \eta \rho^{i_k}$. Specifically, this step length is computed based on the following procedure, where $i_k = \{i : i = 0, 1, \dots\}$ is the smallest non-negative integer i that satisfies the following inequality:

$$-\langle \theta(a_k + \eta \rho^i d_k), d_k \rangle \geq \sigma \eta \rho^i \|\theta(a_k + \eta \rho^i d_k)\| \|d_k\|^2, \quad (2.7)$$

where $\eta > 0$, $\rho \in (0, 1)$, and $\sigma > 0$ are algorithmic parameters.

Furthermore, the projection operator $P_\Theta[\cdot]$ is a critical component that ensures the iterative points remain within the feasible region Θ . Specifically, the projection of a point $a \in \mathbb{R}^n$ onto the set Θ is defined as

$$P_\Theta[a] = \arg \min\{\|a - b\| : b \in \Theta\}, \quad a \in \mathbb{R}^n.$$

This operator identifies the point in Θ closest to a in the Euclidean norm. Moreover, the projection operator is non-expansive, meaning it satisfies the property:

$$\|P_\Theta[a] - P_\Theta[b]\| \leq \|a - b\|. \quad (2.8)$$

Finally, the iterative update rule forms the core of the proposed algorithm, indicating how the next iterative point a_{k+1} is computed from the current iterative point a_k . Specifically, the update is performed by using the following formula:

$$a_{k+1} = P_\Theta[a_k - \gamma w_k \theta(z_k)], \quad w_k = \frac{\langle \theta(z_k), a_k - z_k \rangle}{\|\theta(z_k)\|^2}, \quad (2.9)$$

where $z_k = a_k + t_k d_k$ and $\gamma \in (0, 2)$. This projection-based update ensures that the new iterative point remains feasible and moves towards reducing the objective function.

With the foundational components described above, we now present the detailed steps of an improved LS-RMIL-type conjugate gradient projection algorithm (Abbr. ILR algorithm), which is described as Algorithm 1.

3. Properties analysis

In this section, we provide a comprehensive analysis of the global convergence properties of the ILR algorithm. To facilitate this analysis, we introduce the following key assumptions.

Assumption B:

- (B1) The solution set Θ_* of problem (1.1) is non-empty.
- (B2) The function $\theta(a)$ exhibits a monotonicity property, i.e.,

$$\langle \theta(a) - \theta(b), a - b \rangle \geq 0, \quad \forall a, b \in \mathbb{R}^n.$$

Algorithm 1 An improved LS-RMIL-type conjugate gradient projection algorithm

```

1: Initialization:  $a_0 \in \mathbb{R}^n$ ,  $\mu > 0$ ,  $\tilde{\nu} \in (0, 1)$ ,  $\eta, \sigma > 0$ ,  $\rho \in (0, 1)$ ,  $\tau > 0$ , and set  $k := 0$ .
2: while  $\|\theta_k\| > \tau$  do
3:   Evaluate two parameters  $\beta_k$  and  $\varpi_k$  from (2.2) and search direction  $d_k$  from (2.1).
4:   Evaluate the step length  $t_k$  from (2.7) and set the trial point  $z_k = a_k + t_k d_k$ .
5:   if  $z_k \in \Theta$  and  $\|\theta(z_k)\| < \tau$  then
6:     break.
7:   else
8:     Evaluate the next iterative point  $a_{k+1}$  from (2.9).
9:   end if
10:  Set  $k := k + 1$ .
11: end while

```

These assumptions are fundamental in establishing the convergence behavior of the ILR algorithm as they ensure that the iterative process converges to a solution within the feasible region of problem (1.1).

The following lemma demonstrates that the line search approach defined in (2.7) of the ILR algorithm is indeed well-defined and can be successfully applied in the iterative process.

Lemma 3. *Consider the sequence $\{t_k\}$ generated by the ILR algorithm. Then, there exists a step length t_k at each iteration that satisfies the line search approach defined in (2.7).*

Proof. We proceed by contradiction and assume that inequality (2.7) does not hold. Specifically, suppose there exists a positive index k_0 such that, for all $i \in \{0\} \cup \mathbb{N}$, the following inequality is satisfied:

$$-\langle \theta(a_{k_0} + \eta \rho^i d_{k_0}), d_{k_0} \rangle < \sigma \eta \rho^i \|\theta(a_{k_0} + \eta \rho^i d_{k_0})\| \|d_{k_0}\|^2.$$

By utilizing the continuity of θ and taking the limit as $i \rightarrow \infty$, the above inequality yields:

$$-\langle \theta(a_{k_0}), d_{k_0} \rangle \leq 0. \quad (3.1)$$

On the other hand, invoking Lemma 1 and again taking the limit as $i \rightarrow \infty$, we obtain:

$$-\langle \theta(a_{k_0}), d_{k_0} \rangle \geq M \|\theta(a_{k_0})\|^2 > 0,$$

which clearly contradicts inequality (3.1). This contradiction implies that the initial assumption must be false, and therefore inequality (2.7) must hold. \square

The following lemma establishes that the sequence $\{a_k\}$ generated by the ILR algorithm exhibits monotonic behavior with respect to the solutions set Θ_* of problem (1.1).

Lemma 4. *Consider the sequences $\{a_k\}$ and $\{z_k\}$ generated by the ILR algorithm. Then, the following properties hold:*

- (i) *The sequence $\{a_k\}$ is bounded, meaning that there exists a constant $D > 0$ such that $\|a_k\| \leq D$ for all $k \geq 0$.*
- (ii) *The sequence $\{z_k\}$ converges to the sequence $\{a_k\}$, i.e., $\lim_{k \rightarrow \infty} \|z_k - a_k\| = 0$.*

Proof. From the definition of the projection operator $P_{\Theta}[\cdot]$ and the non-expensive property defined in (2.8), we can derive the following inequality:

$$\begin{aligned}\|a_{k+1} - a_*\|^2 &= \|P_{\Theta}[a_k - \gamma w_k \theta(z_k)] - P_{\Theta}[a_*]\|^2 \\ &\leq \|a_k - \gamma w_k \theta(z_k) - a_*\|^2 \\ &= \|a_k - a_*\|^2 - 2\gamma w_k \langle \theta(z_k), a_k - a_* \rangle + \gamma^2 w_k^2 \|\theta(z_k)\|^2,\end{aligned}\quad (3.2)$$

where a_* denotes a solution of problem (1.1). Next, starting from Assumption B2, the definition of z_k , and the line search approach (2.7), we can further establish the following inequality:

$$\begin{aligned}\langle \theta(z_k), a_k - a_* \rangle &= \langle \theta(z_k), a_k - z_k \rangle + \langle \theta(z_k), z_k - a_* \rangle - \langle \theta(a_*), z_k - a_* \rangle \\ &\geq \langle \theta(z_k), a_k - z_k \rangle \\ &\geq \sigma t_k^2 \|\theta(z_k)\| \|d_k\|^2.\end{aligned}\quad (3.3)$$

Combining with (3.2), (3.3), and the definition of w_k , we can derive

$$\begin{aligned}\|a_{k+1} - a_*\|^2 &\leq \|a_k - a_*\|^2 - 2\gamma w_k \langle \theta(z_k), a_k - z_k \rangle + \gamma^2 w_k^2 \|\theta(z_k)\|^2, \\ &= \|a_k - a_*\|^2 - 2\gamma w_k^2 \|\theta(z_k)\|^2 + \gamma^2 w_k^2 \|\theta(z_k)\|^2, \\ &= \|a_k - a_*\|^2 - (2\gamma - \gamma^2) w_k^2 \|\theta(z_k)\|^2,\end{aligned}\quad (3.4)$$

Given the definition of w_k and (3.3), we have

$$\|\theta(z_k)\|^2 w_k = \langle \theta(z_k), a_k - z_k \rangle \geq \sigma t_k^2 \|\theta(z_k)\| \|d_k\|^2,$$

which implies that $\|\theta(z_k)\| w_k \geq \sigma t_k^2 \|d_k\|^2$. Substituting this into (3.4), we obtain

$$\begin{aligned}\|a_{k+1} - a_*\|^2 &\leq \|a_k - a_*\|^2 - (2\gamma - \gamma^2) (\sigma t_k^2 \|d_k\|^2)^2, \\ &= \|a_k - a_*\|^2 - (2\gamma - \gamma^2) \sigma^2 t_k^4 \|d_k\|^4, \\ &= \|a_k - a_*\|^2 - (2\gamma - \gamma^2) \sigma^2 \|a_k - z_k\|^4.\end{aligned}\quad (3.5)$$

This result indicates that the sequence $\{\|a_k - a_*\|\}$ is monotonically decreasing, meaning that it consistently reduces as k increases. Hence, the sequence $\{a_k\}$ is bounded.

By reorganizing the formula defined in (3.5), we obtain

$$\begin{aligned}(2\gamma - \gamma^2) \sigma^2 \sum_{k=0}^{\infty} \|a_k - z_k\|^4 &\leq \sum_{k=0}^{\infty} (\|a_k - a_*\|^2 - \|a_{k+1} - a_*\|^2) \\ &\leq \|a_0 - a_*\|.\end{aligned}\quad (3.6)$$

This implies that $\lim_{k \rightarrow \infty} \|z_k - a_k\| = 0$. □

Theorem 1. Consider the sequence $\{\theta_k\}$ generated by the ILR algorithm. Then, the following conclusion is satisfied:

$$\lim_{k \rightarrow \infty} \|\theta_k\| = 0. \quad (3.7)$$

Proof. To demonstrate the desired result, we begin by assuming the contrary. Suppose that there exists a constant $A_1 > 0$ such that $\|\theta_k\| > A_1$ for all $k \geq 0$. This assumption, combined with Lemma 2, gives us the following relation $\|d_k\| \geq M \|\theta_k\| > M A_1$ for all $k \geq 0$. Given the continuity of the function $\theta(a)$ and the boundedness of the sequence $\{a_k\}$, it follows that the sequence $\{\theta_k\}$ is also bounded. In other

words, there exists a non-negative constant A_2 such that $\|\theta_k\| \leq A_2$ for all $k \geq 0$. By incorporating this bound with Lemma 2, we obtain $\|d_k\| \leq N\|\theta_k\| \leq NA_2$ for all $k \geq 0$. The two inequalities derived above imply that the sequence $\{d_k\}$ is bounded. Together with Lemma 4(ii) and the definition of z_k , we have $\lim_{k \rightarrow \infty} \|z_k - a_k\| = \lim_{k \rightarrow \infty} \|a_k + t_k d_k - a_k\| = \lim_{k \rightarrow \infty} t_k \|d_k\| = 0$, which leads to the conclusion that $\lim_{k \rightarrow \infty} t_k = 0$ with the boundedness of the sequence $\{d_k\}$.

Since the sequences $\{a_k\}$ and $\{d_k\}$ are both bounded, we can extract two convergent subsequences, $\{a_{k_i}\}$ and $\{d_{k_i}\}$, such that $\lim_{i \rightarrow \infty, i \in \mathcal{K}} a_{k_i} = \bar{a}$ and $\lim_{i \rightarrow \infty, i \in \mathcal{K}} d_{k_i} = \bar{d}$, where \mathcal{K} denotes an infinite index set. Utilizing Lemma 1, we have $-\langle \theta_{k_i}, d_{k_i} \rangle \geq M\|\theta_{k_i}\|^2$. Taking the limit as $i \rightarrow \infty$ in the above inequality and invoking the continuity of $\theta(a)$, we obtain

$$-\langle \theta(\bar{a}), \bar{d} \rangle \geq M\|\theta(\bar{a})\|^2 \geq MA_1^2 > 0.$$

Furthermore, we adopt the line search approach defined in (2.7), which implies the following inequality holds: $-\langle \theta(a_{k_i} + (\eta\rho)^{-1}t_{k_i}d_{k_i}), d_{k_i} \rangle < \sigma\eta(\eta\rho)^{-1}t_{k_i}\|\theta(a_{k_i} + (\eta\rho)^{-1}t_{k_i}d_{k_i})\|\|d_{k_i}\|^2$. Taking the limit as $i \rightarrow \infty$ in the above inequality, and using the continuity of $\theta(a)$, we conclude

$$-\langle \theta(\bar{a}), \bar{d} \rangle \leq 0.$$

These two results directly contradict each other. Therefore, the assumption that $\|\theta_k\| > A_1$ for all $k \geq 0$ must be false, and the desired result follows. \square

4. Numerical experiments for systems of nonlinear equations

In this section, we evaluate the effectiveness of the proposed ILR algorithm through a comprehensive set of numerical experiments. These experiments are designed to solve large-scale systems of nonlinear equations with convex constraints, thereby assessing the algorithm's computational efficiency. For benchmarking purposes, we compare the ILR algorithm with two established methods (e.g., VRMILP and DFPRPMHS) across various test problems, initial points, and dimensional settings.

4.1. General setups

In this section, we utilize the ILR algorithm to address large-scale systems of nonlinear equations with convex constraints. We then compare it with two existing algorithms: the VRMILP algorithm [26] and the DFPRPMHS algorithm [27]. All experimental codes are executed on a 64-bit Ubuntu 20.04.2 LTS operating system with an Intel(R) Xeon(R) Gold 5115 2.40GHz CPU. The parameters for the ILR algorithm are set as follows:

$$\mu = 0.02, \quad \tilde{\nu} = 0.105, \quad \eta = 1, \quad \sigma = 10^{-4}, \quad \rho = 0.74, \quad \tau = 10^{-5}.$$

For the VRMILP and DFPRPMHS algorithms, we adhere to the parameter settings provided in their respective original works. Seven test problems are selected for evaluation, with problem dimensions set at $\{5,000 \ 10,000 \ 50,000 \ 100,000 \ 150,000\}$. Each test problem is initialized by the following points: $a_1 = (\frac{1}{2}, \frac{1}{2^2}, \dots, \frac{1}{2^n})$, $a_2 = (0, \frac{1}{n}, \frac{2}{n}, \dots, \frac{n-1}{n})$, $a_3 = (1, \frac{1}{2}, \dots, \frac{1}{n})$, $a_4 = (\frac{1}{n}, \frac{2}{n}, \dots, \frac{n}{n})$, $a_5 = (\frac{1}{3}, \frac{1}{3^2}, \dots, \frac{1}{3^n})$, $a_6 = (2, 2, \dots, 2)$, $a_7 = (1 - \frac{1}{n}, 1 - \frac{2}{n}, \dots, 1 - \frac{n}{n})$, $a_8 \in [0, 1]^n$. The stopping criteria for all algorithms

is set to either $\theta_k \leq \tau$ or a maximum of 3000 iterations. Here, $\theta(a) = (\theta_1(a), \theta_2(a), \dots, \theta_n(a))^T$ with $a = (a_1, a_2, \dots, a_n)^T$. The seven test problems are described as follows:

Problem 1 [7]:

$$\theta_i(a) = e^{a_i} - 1, \quad \text{for } i = 1, 2, \dots, n,$$

with the constraint set $\Theta = \mathbb{R}_+^n$. The unique solution is $a_* = (0, 0, \dots, 0)^T$.

Problem 2 [7]:

$$\theta_i(a) = \frac{i}{n} e^{a_i} - 1, \quad \text{for } i = 1, 2, \dots, n,$$

with the constraint set $\Theta = \mathbb{R}_+^n$.

Problem 3 [5]:

$$\theta_i(a) = \log(a_i + 1) - \frac{a_i}{n}, \quad \text{for } i = 1, 2, \dots, n,$$

with the constraint set $\Theta = [-1, +\infty)$.

Problem 4 [5]:

$$\theta_i(a) = (e^{a_i})^2 + 3 \sin(a_i) \cos(a_i) - 1, \quad \text{for } i = 1, 2, \dots, n,$$

with the constraint set $\Theta = \mathbb{R}_+^n$.

Problem 5 [5]:

$$\begin{aligned} \theta_1(a) &= 2a_1 + \sin(a_1) - 1, \\ \theta_i(a) &= 2a_{i-1} + 2a_i + \sin(a_i) - 1, \quad \text{for } i = 2, 3, \dots, n-1, \\ \theta_n(a) &= 2a_n + \sin(a_n) - 1, \end{aligned}$$

with the constraint set $\Theta = \mathbb{R}_+^n$.

Problem 6 [7]:

$$\theta_i(a) = \frac{1}{n} e^{a_i} - 1, \quad \text{for } i = 1, 2, \dots, n,$$

with the constraint set $\Theta = \mathbb{R}_+^n$.

Problem 7 [5]:

$$\theta_i(a) = a_i - 2 \sin(|a_i - 1|), \quad \text{for } i = 1, 2, \dots, n,$$

with the constraint set $\Theta = \mathbb{R}_+^n$.

4.2. Numerical reports

The performance of the ILR, VRMILP, and DFPRPMHS algorithms are systematically evaluated through a series of test problems, with the numerical results presented in Tables 1–7. In these tables, “Init” refers to the initial point used in each test problem, “ n ” refers to the problem dimension multiplied by 1000, “CPUT” refers to the CPU time in seconds, “Nfunc” refers to the number of function evaluations, and “Niter” refers to the number of iterations. A notable observation from the numerical results is that all three algorithms successfully solve the test problem across various initial points and problem dimensions. To be specific, the ILR algorithm demonstrates superior performance in most cases compared to the other two algorithms.

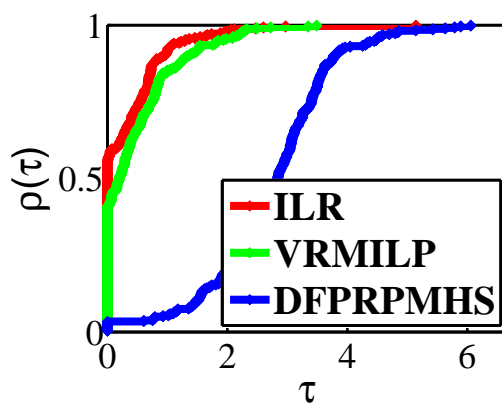


Figure 1. Performance profiles on CPUT.

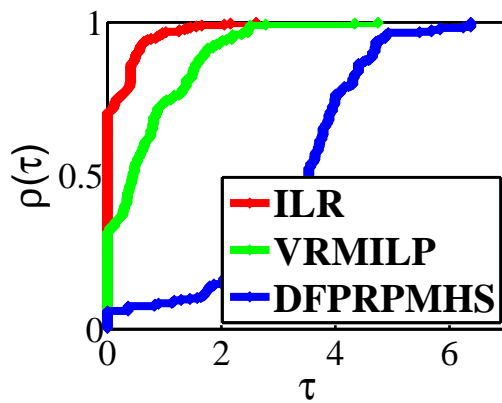


Figure 2. Performance profiles on Nfunc.

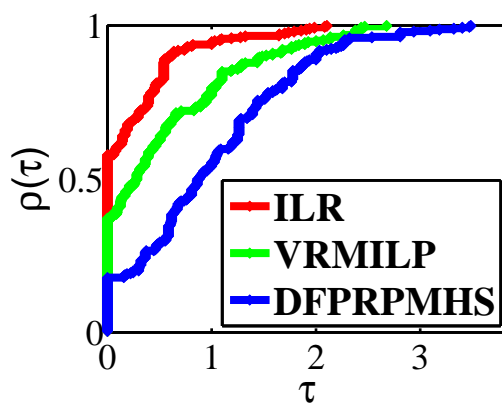


Figure 3. Performance profiles on Niter.

Table 1. Numerical results for Problem 1.

Inti(n)	ILR	VRMILP	DFPRPMHS
	CPUT/Nfunc/Niter	CPUT/Nfunc/Niter	CPUT/Nfunc/Niter
$a_1(5)$	$1.93 \times 10^{-3}/4/1$	$4.45 \times 10^{-4}/4/1$	$1.57 \times 10^{-2}/332/7$
$a_2(5)$	$1.84 \times 10^{-3}/23/7$	$3.47 \times 10^{-3}/60/15$	$2.15 \times 10^{-2}/487/24$
$a_3(5)$	$2.11 \times 10^{-3}/29/9$	$2.16 \times 10^{-3}/37/8$	$1.75 \times 10^{-2}/447/10$
$a_4(5)$	$1.55 \times 10^{-3}/23/7$	$2.13 \times 10^{-3}/39/9$	$1.86 \times 10^{-2}/405/31$
$a_5(5)$	$1.57 \times 10^{-3}/23/7$	$3.33 \times 10^{-3}/60/15$	$2.04 \times 10^{-2}/487/24$
$a_6(5)$	$5.08 \times 10^{-4}/7/1$	$4.16 \times 10^{-4}/7/1$	$2.00 \times 10^{-3}/9/1$
$a_7(5)$	$1.92 \times 10^{-3}/23/7$	$3.46 \times 10^{-3}/60/15$	$2.07 \times 10^{-2}/487/24$
$a_8(5)$	$2.21 \times 10^{-3}/30/9$	$2.16 \times 10^{-3}/39/9$	$2.30 \times 10^{-2}/563/14$
$a_1(10)$	$6.36 \times 10^{-4}/4/1$	$4.22 \times 10^{-4}/4/1$	$2.50 \times 10^{-2}/332/7$
$a_2(10)$	$3.69 \times 10^{-3}/23/7$	$6.19 \times 10^{-3}/57/14$	$4.11 \times 10^{-2}/487/24$
$a_3(10)$	$4.66 \times 10^{-3}/29/9$	$3.96 \times 10^{-3}/37/8$	$3.44 \times 10^{-2}/447/10$
$a_4(10)$	$3.48 \times 10^{-3}/23/7$	$4.12 \times 10^{-3}/39/9$	$3.59 \times 10^{-2}/408/32$
$a_5(10)$	$3.45 \times 10^{-3}/23/7$	$6.12 \times 10^{-3}/57/14$	$3.98 \times 10^{-2}/487/24$
$a_6(10)$	$8.35 \times 10^{-4}/7/1$	$7.34 \times 10^{-4}/7/1$	$1.99 \times 10^{-3}/9/1$
$a_7(10)$	$3.56 \times 10^{-3}/23/7$	$6.10 \times 10^{-3}/57/14$	$3.88 \times 10^{-2}/487/24$
$a_8(10)$	$6.37 \times 10^{-3}/41/13$	$5.28 \times 10^{-3}/48/12$	$4.40 \times 10^{-2}/563/14$
$a_1(50)$	$3.75 \times 10^{-3}/4/1$	$1.64 \times 10^{-3}/4/1$	$9.06 \times 10^{-2}/332/7$
$a_2(50)$	$1.35 \times 10^{-2}/23/7$	$2.48 \times 10^{-2}/60/15$	$1.51 \times 10^{-1}/596/26$
$a_3(50)$	$1.27 \times 10^{-2}/29/9$	$1.72 \times 10^{-2}/37/8$	$1.02 \times 10^{-1}/447/10$
$a_4(50)$	$9.17 \times 10^{-3}/23/7$	$1.11 \times 10^{-2}/39/9$	$1.39 \times 10^{-1}/602/28$
$a_5(50)$	$1.39 \times 10^{-2}/23/7$	$2.47 \times 10^{-2}/60/15$	$1.49 \times 10^{-1}/596/26$
$a_6(50)$	$2.67 \times 10^{-3}/7/1$	$4.30 \times 10^{-3}/7/1$	$5.98 \times 10^{-3}/9/1$
$a_7(50)$	$1.13 \times 10^{-2}/23/7$	$1.95 \times 10^{-2}/60/15$	$1.32 \times 10^{-1}/596/26$
$a_8(50)$	$1.06 \times 10^{-2}/23/7$	$1.86 \times 10^{-2}/56/14$	$1.30 \times 10^{-1}/576/16$
$a_1(100)$	$7.76 \times 10^{-3}/4/1$	$2.44 \times 10^{-3}/4/1$	$1.21 \times 10^{-1}/332/7$
$a_2(100)$	$1.49 \times 10^{-2}/23/7$	$3.51 \times 10^{-2}/66/17$	$1.44 \times 10^{-1}/487/24$
$a_3(100)$	$1.69 \times 10^{-2}/29/9$	$1.64 \times 10^{-2}/37/8$	$1.24 \times 10^{-1}/447/10$
$a_4(100)$	$1.49 \times 10^{-2}/23/7$	$1.56 \times 10^{-2}/39/9$	$1.84 \times 10^{-1}/602/28$
$a_5(100)$	$1.24 \times 10^{-2}/23/7$	$2.76 \times 10^{-2}/66/17$	$1.38 \times 10^{-1}/487/24$
$a_6(100)$	$2.91 \times 10^{-3}/7/1$	$2.55 \times 10^{-3}/7/1$	$5.45 \times 10^{-3}/9/1$
$a_7(100)$	$1.22 \times 10^{-2}/23/7$	$2.67 \times 10^{-2}/66/17$	$1.50 \times 10^{-1}/487/24$
$a_8(100)$	$1.32 \times 10^{-2}/23/7$	$3.29 \times 10^{-2}/60/15$	$1.92 \times 10^{-1}/677/18$
$a_1(150)$	$3.49 \times 10^{-3}/4/1$	$2.63 \times 10^{-3}/4/1$	$1.13 \times 10^{-1}/332/7$
$a_2(150)$	$1.88 \times 10^{-2}/23/7$	$3.88 \times 10^{-2}/75/19$	$2.84 \times 10^{-1}/602/28$
$a_3(150)$	$1.95 \times 10^{-2}/29/9$	$1.88 \times 10^{-2}/37/8$	$1.71 \times 10^{-1}/447/10$
$a_4(150)$	$1.65 \times 10^{-2}/23/7$	$2.08 \times 10^{-2}/39/9$	$2.36 \times 10^{-1}/602/28$
$a_5(150)$	$1.65 \times 10^{-2}/23/7$	$3.80 \times 10^{-2}/75/19$	$2.40 \times 10^{-1}/602/28$
$a_6(150)$	$5.11 \times 10^{-3}/7/1$	$4.06 \times 10^{-3}/7/1$	$1.05 \times 10^{-2}/9/1$
$a_7(150)$	$1.75 \times 10^{-2}/23/7$	$4.80 \times 10^{-2}/75/19$	$2.46 \times 10^{-1}/602/28$
$a_8(150)$	$1.95 \times 10^{-2}/23/7$	$2.97 \times 10^{-2}/45/11$	$2.34 \times 10^{-1}/575/18$

Table 2. Numerical results for Problem 2.

Inti(n)	ILR	VRMILP	DFPRPMHS
	CPUT/Nfunc/Niter	CPUT/Nfunc/Niter	CPUT/Nfunc/Niter
$a_1(5)$	$8.86 \times 10^{-3}/\mathbf{29/11}$	$\mathbf{8.84} \times 10^{-3}/96/25$	$3.91 \times 10^{-2}/576/23$
$a_2(5)$	$2.64 \times 10^{-2}/400/52$	$\mathbf{1.73} \times 10^{-2}/\mathbf{281/31}$	$5.06 \times 10^{-2}/893/43$
$a_3(5)$	$\mathbf{2.85} \times 10^{-3}/\mathbf{30/11}$	$6.36 \times 10^{-3}/88/23$	$2.04 \times 10^{-2}/300/34$
$a_4(5)$	$2.69 \times 10^{-2}/414/53$	$\mathbf{1.86} \times 10^{-2}/\mathbf{310/30}$	$4.14 \times 10^{-2}/692/44$
$a_5(5)$	$\mathbf{4.95} \times 10^{-3}/\mathbf{54/20}$	$7.76 \times 10^{-3}/107/28$	$2.56 \times 10^{-2}/406/35$
$a_6(5)$	$1.57 \times 10^{-2}/232/31$	$\mathbf{7.92} \times 10^{-3}/\mathbf{114/25}$	$5.95 \times 10^{-2}/1056/34$
$a_7(5)$	$\mathbf{5.33} \times 10^{-3}/\mathbf{54/20}$	$7.78 \times 10^{-3}/107/28$	$2.59 \times 10^{-2}/406/35$
$a_8(5)$	$4.65 \times 10^{-2}/727/76$	$\mathbf{1.90} \times 10^{-2}/\mathbf{305/36}$	$4.12 \times 10^{-2}/688/41$
$a_1(10)$	$\mathbf{1.22} \times 10^{-2}/\mathbf{56/20}$	$1.26 \times 10^{-2}/84/22$	$5.03 \times 10^{-2}/411/37$
$a_2(10)$	$5.78 \times 10^{-2}/435/47$	$\mathbf{4.50} \times 10^{-2}/\mathbf{398/34}$	$7.79 \times 10^{-2}/703/\mathbf{22}$
$a_3(10)$	$\mathbf{6.49} \times 10^{-3}/\mathbf{29/11}$	$1.56 \times 10^{-2}/82/22$	$5.64 \times 10^{-2}/353/55$
$a_4(10)$	$8.59 \times 10^{-2}/582/57$	$\mathbf{4.34} \times 10^{-2}/\mathbf{387/36}$	$8.26 \times 10^{-2}/721/\mathbf{25}$
$a_5(10)$	$\mathbf{1.05} \times 10^{-2}/\mathbf{50/18}$	$1.92 \times 10^{-2}/128/32$	$4.28 \times 10^{-2}/300/37$
$a_6(10)$	$3.89 \times 10^{-2}/276/35$	$\mathbf{1.51} \times 10^{-2}/\mathbf{108/24}$	$9.90 \times 10^{-2}/851/40$
$a_7(10)$	$\mathbf{1.07} \times 10^{-2}/\mathbf{50/18}$	$1.87 \times 10^{-2}/128/32$	$4.04 \times 10^{-2}/300/37$
$a_8(10)$	$8.75 \times 10^{-2}/673/60$	$\mathbf{6.39} \times 10^{-2}/\mathbf{541/52}$	$1.09 \times 10^{-1}/966/\mathbf{39}$
$a_1(50)$	$5.65 \times 10^{-2}/\mathbf{57/20}$	$\mathbf{5.36} \times 10^{-2}/90/24$	$1.69 \times 10^{-1}/320/41$
$a_2(50)$	$4.89 \times 10^{-1}/997/80$	$4.34 \times 10^{-1}/951/57$	$\mathbf{3.28} \times 10^{-1}/\mathbf{731/24}$
$a_3(50)$	$\mathbf{2.44} \times 10^{-2}/\mathbf{35/13}$	$5.55 \times 10^{-2}/99/26$	$1.42 \times 10^{-1}/327/35$
$a_4(50)$	$4.42 \times 10^{-1}/926/83$	$3.80 \times 10^{-1}/957/55$	$\mathbf{3.09} \times 10^{-1}/\mathbf{731/24}$
$a_5(50)$	$\mathbf{4.08} \times 10^{-2}/\mathbf{55/19}$	$6.92 \times 10^{-2}/124/31$	$1.93 \times 10^{-1}/409/36$
$a_6(50)$	$2.48 \times 10^{-1}/510/51$	$\mathbf{5.96} \times 10^{-2}/\mathbf{114/25}$	$4.73 \times 10^{-1}/1119/36$
$a_7(50)$	$\mathbf{3.77} \times 10^{-2}/\mathbf{55/19}$	$6.56 \times 10^{-2}/124/31$	$1.99 \times 10^{-1}/409/36$
$a_8(50)$	$4.88 \times 10^{-1}/1011/86$	$3.19 \times 10^{-1}/699/50$	$\mathbf{3.04} \times 10^{-1}/\mathbf{686/20}$
$a_1(100)$	$7.76 \times 10^{-2}/\mathbf{75/27}$	$\mathbf{7.69} \times 10^{-2}/92/\mathbf{24}$	$4.11 \times 10^{-1}/613/31$
$a_2(100)$	$8.14 \times 10^{-1}/1187/92$	$8.34 \times 10^{-1}/1259/65$	$\mathbf{4.43} \times 10^{-1}/\mathbf{727/24}$
$a_3(100)$	$9.47 \times 10^{-2}/96/33$	$\mathbf{7.82} \times 10^{-2}/\mathbf{94/25}$	$2.83 \times 10^{-1}/413/39$
$a_4(100)$	$7.69 \times 10^{-1}/1073/90$	$8.34 \times 10^{-1}/1193/66$	$\mathbf{5.16} \times 10^{-1}/\mathbf{728/24}$
$a_5(100)$	$\mathbf{9.05} \times 10^{-2}/\mathbf{76/26}$	$1.01 \times 10^{-1}/119/31$	$3.28 \times 10^{-1}/493/29$
$a_6(100)$	$3.62 \times 10^{-1}/505/46$	$\mathbf{9.80} \times 10^{-2}/\mathbf{122/27}$	$6.84 \times 10^{-1}/1006/46$
$a_7(100)$	$\mathbf{8.67} \times 10^{-2}/\mathbf{76/26}$	$9.83 \times 10^{-2}/119/31$	$3.27 \times 10^{-1}/493/29$
$a_8(100)$	$1.18 \times 10^0/1610/106$	$5.32 \times 10^{-1}/800/52$	$\mathbf{4.04} \times 10^{-1}/\mathbf{608/25}$
$a_1(150)$	$\mathbf{8.00} \times 10^{-2}/\mathbf{54/20}$	$1.15 \times 10^{-1}/99/26$	$3.48 \times 10^{-1}/330/40$
$a_2(150)$	$1.15 \times 10^0/1130/87$	$1.34 \times 10^0/1408/76$	$\mathbf{7.09} \times 10^{-1}/\mathbf{726/24}$
$a_3(150)$	$\mathbf{8.44} \times 10^{-2}/\mathbf{57/18}$	$1.25 \times 10^{-1}/102/26$	$4.08 \times 10^{-1}/400/35$
$a_4(150)$	$1.17 \times 10^0/1165/95$	$1.26 \times 10^0/1365/69$	$\mathbf{6.80} \times 10^{-1}/\mathbf{726/24}$
$a_5(150)$	$\mathbf{9.90} \times 10^{-2}/\mathbf{69/25}$	$1.44 \times 10^{-1}/124/32$	$4.11 \times 10^{-1}/407/38$
$a_6(150)$	$7.33 \times 10^{-1}/714/63$	$\mathbf{1.32} \times 10^{-1}/\mathbf{117/26}$	$8.80 \times 10^0/8853/275$
$a_7(150)$	$\mathbf{1.24} \times 10^{-1}/\mathbf{69/25}$	$1.73 \times 10^{-1}/124/32$	$4.47 \times 10^{-1}/407/38$
$a_8(150)$	$1.33 \times 10^0/1224/91$	$1.27 \times 10^0/1255/71$	$\mathbf{7.01} \times 10^{-1}/\mathbf{657/29}$

Table 3. Numerical results for Problem 3.

Inti(n)	ILR	VRMILP	DFPRPMHS
	CPUT/Nfunc/Niter	CPUT/Nfunc/Niter	CPUT/Nfunc/Niter
$a_1(5)$	$8.33 \times 10^{-3}/23/8$	$2.50 \times 10^{-3}/21/9$	$3.90 \times 10^{-2}/544/11$
$a_2(5)$	$8.23 \times 10^{-3}/61/21$	$8.17 \times 10^{-3}/82/22$	$6.76 \times 10^{-2}/977/19$
$a_3(5)$	$5.10 \times 10^{-3}/45/16$	$5.06 \times 10^{-3}/60/16$	$3.11 \times 10^{-2}/492/30$
$a_4(5)$	$7.16 \times 10^{-3}/66/23$	$6.87 \times 10^{-3}/82/22$	$5.68 \times 10^{-2}/977/19$
$a_5(5)$	$6.90 \times 10^{-3}/61/21$	$7.31 \times 10^{-3}/82/22$	$5.94 \times 10^{-2}/977/19$
$a_6(5)$	$2.28 \times 10^{-3}/19/7$	$3.53 \times 10^{-3}/33/14$	$6.56 \times 10^{-2}/1084/21$
$a_7(5)$	$7.05 \times 10^{-3}/61/21$	$6.94 \times 10^{-3}/82/22$	$5.87 \times 10^{-2}/977/19$
$a_8(5)$	$4.85 \times 10^{-3}/44/15$	$8.76 \times 10^{-3}/105/28$	$5.83 \times 10^{-2}/977/19$
$a_1(10)$	$6.23 \times 10^{-3}/23/8$	$3.88 \times 10^{-3}/21/9$	$5.47 \times 10^{-2}/544/11$
$a_2(10)$	$9.75 \times 10^{-3}/49/17$	$1.26 \times 10^{-2}/85/23$	$1.05 \times 10^{-1}/986/21$
$a_3(10)$	$1.12 \times 10^{-2}/54/19$	$8.97 \times 10^{-3}/60/16$	$5.92 \times 10^{-2}/499/31$
$a_4(10)$	$1.67 \times 10^{-2}/78/28$	$1.27 \times 10^{-2}/85/23$	$1.07 \times 10^{-1}/986/21$
$a_5(10)$	$1.04 \times 10^{-2}/49/17$	$1.24 \times 10^{-2}/85/23$	$1.09 \times 10^{-1}/986/21$
$a_6(10)$	$4.08 \times 10^{-3}/19/7$	$5.89 \times 10^{-3}/33/14$	$1.06 \times 10^{-1}/977/19$
$a_7(10)$	$1.10 \times 10^{-2}/49/17$	$1.44 \times 10^{-2}/85/23$	$1.11 \times 10^{-1}/986/21$
$a_8(10)$	$8.57 \times 10^{-3}/44/15$	$1.47 \times 10^{-2}/98/27$	$9.76 \times 10^{-2}/890/24$
$a_1(50)$	$2.84 \times 10^{-2}/23/8$	$1.39 \times 10^{-2}/21/9$	$1.18 \times 10^{-1}/437/9$
$a_2(50)$	$4.56 \times 10^{-2}/74/26$	$3.72 \times 10^{-2}/94/25$	$3.08 \times 10^{-1}/978/19$
$a_3(50)$	$2.33 \times 10^{-2}/44/15$	$2.67 \times 10^{-2}/60/16$	$1.04 \times 10^{-1}/279/31$
$a_4(50)$	$4.24 \times 10^{-2}/61/21$	$4.43 \times 10^{-2}/93/25$	$2.94 \times 10^{-1}/978/19$
$a_5(50)$	$5.26 \times 10^{-2}/74/26$	$4.02 \times 10^{-2}/94/25$	$2.85 \times 10^{-1}/978/19$
$a_6(50)$	$1.01 \times 10^{-2}/19/7$	$1.48 \times 10^{-2}/35/15$	$2.93 \times 10^{-1}/977/19$
$a_7(50)$	$4.55 \times 10^{-2}/74/26$	$4.41 \times 10^{-2}/94/25$	$2.97 \times 10^{-1}/978/19$
$a_8(50)$	$4.27 \times 10^{-2}/63/22$	$5.60 \times 10^{-2}/102/27$	$2.92 \times 10^{-1}/978/19$
$a_1(100)$	$1.51 \times 10^{-2}/20/7$	$1.14 \times 10^{-2}/18/7$	$1.52 \times 10^{-1}/437/9$
$a_2(100)$	$5.81 \times 10^{-2}/72/25$	$6.66 \times 10^{-2}/104/28$	$3.86 \times 10^{-1}/978/19$
$a_3(100)$	$4.40 \times 10^{-2}/58/20$	$3.25 \times 10^{-2}/60/16$	$1.36 \times 10^{-1}/364/19$
$a_4(100)$	$4.69 \times 10^{-2}/61/21$	$5.44 \times 10^{-2}/104/28$	$4.02 \times 10^{-1}/978/19$
$a_5(100)$	$5.39 \times 10^{-2}/72/25$	$5.64 \times 10^{-2}/104/28$	$3.98 \times 10^{-1}/978/19$
$a_6(100)$	$1.69 \times 10^{-2}/19/7$	$2.70 \times 10^{-2}/35/15$	$4.70 \times 10^{-1}/1085/21$
$a_7(100)$	$5.06 \times 10^{-2}/72/25$	$5.60 \times 10^{-2}/104/28$	$4.14 \times 10^{-1}/978/19$
$a_8(100)$	$6.54 \times 10^{-2}/76/26$	$5.91 \times 10^{-2}/109/29$	$3.98 \times 10^{-1}/978/19$
$a_1(150)$	$1.99 \times 10^{-2}/20/7$	$1.50 \times 10^{-2}/18/7$	$2.04 \times 10^{-1}/437/9$
$a_2(150)$	$7.71 \times 10^{-2}/71/25$	$8.53 \times 10^{-2}/99/27$	$5.09 \times 10^{-1}/978/19$
$a_3(150)$	$5.08 \times 10^{-2}/51/17$	$4.93 \times 10^{-2}/60/16$	$1.65 \times 10^{-1}/274/19$
$a_4(150)$	$8.24 \times 10^{-2}/74/25$	$8.47 \times 10^{-2}/99/27$	$5.37 \times 10^{-1}/978/19$
$a_5(150)$	$7.76 \times 10^{-2}/71/25$	$6.93 \times 10^{-2}/99/27$	$5.39 \times 10^{-1}/978/19$
$a_6(150)$	$2.37 \times 10^{-2}/22/8$	$3.91 \times 10^{-2}/35/15$	$5.64 \times 10^{-1}/978/19$
$a_7(150)$	$8.44 \times 10^{-2}/71/25$	$8.50 \times 10^{-2}/99/27$	$5.38 \times 10^{-1}/978/19$
$a_8(150)$	$6.24 \times 10^{-2}/62/22$	$7.60 \times 10^{-2}/105/28$	$5.27 \times 10^{-1}/978/19$

Table 4. Numerical results for Problem 4.

Inti(n)	ILR	VRMILP	DFPRPMHS
	CPUT/Nfunc/Niter	CPUT/Nfunc/Niter	CPUT/Nfunc/Niter
$a_1(5)$	$1.58 \times 10^{-1}/3/1$	$4.47 \times 10^{-3}/3/1$	$2.10 \times 10^{-2}/3/1$
$a_2(5)$	$1.70 \times 10^{-2}/44/5$	$1.91 \times 10^{-2}/149/16$	$4.55 \times 10^{-2}/651/16$
$a_3(5)$	$9.99 \times 10^{-3}/63/8$	$1.86 \times 10^{-2}/247/26$	$2.06 \times 10^{-2}/259/7$
$a_4(5)$	$3.70 \times 10^{-3}/44/5$	$1.33 \times 10^{-2}/225/24$	$2.86 \times 10^{-2}/509/12$
$a_5(5)$	$5.19 \times 10^{-3}/44/5$	$1.36 \times 10^{-2}/149/16$	$3.72 \times 10^{-2}/651/16$
$a_6(5)$	$8.76 \times 10^{-4}/4/1$	$5.51 \times 10^{-4}/4/1$	$2.28 \times 10^{-3}/4/1$
$a_7(5)$	$3.48 \times 10^{-3}/44/5$	$9.75 \times 10^{-3}/149/16$	$3.18 \times 10^{-2}/651/16$
$a_8(5)$	$3.25 \times 10^{-3}/44/5$	$1.24 \times 10^{-2}/200/21$	$3.11 \times 10^{-2}/626/14$
$a_1(10)$	$1.03 \times 10^{-3}/3/1$	$5.12 \times 10^{-4}/3/1$	$3.02 \times 10^{-3}/3/1$
$a_2(10)$	$5.99 \times 10^{-3}/44/5$	$2.83 \times 10^{-2}/243/26$	$6.79 \times 10^{-2}/660/16$
$a_3(10)$	$8.56 \times 10^{-3}/63/8$	$2.97 \times 10^{-2}/246/26$	$2.56 \times 10^{-2}/259/7$
$a_4(10)$	$5.47 \times 10^{-3}/44/5$	$1.59 \times 10^{-2}/141/15$	$5.05 \times 10^{-2}/509/12$
$a_5(10)$	$5.66 \times 10^{-3}/44/5$	$2.84 \times 10^{-2}/243/26$	$6.01 \times 10^{-2}/660/16$
$a_6(10)$	$1.04 \times 10^{-3}/4/1$	$6.07 \times 10^{-4}/4/1$	$1.69 \times 10^{-3}/4/1$
$a_7(10)$	$5.08 \times 10^{-3}/44/5$	$2.48 \times 10^{-2}/243/26$	$5.86 \times 10^{-2}/660/16$
$a_8(10)$	$5.28 \times 10^{-3}/44/5$	$2.45 \times 10^{-2}/229/24$	$3.55 \times 10^{-2}/364/17$
$a_1(50)$	$8.54 \times 10^{-3}/3/1$	$1.09 \times 10^{-3}/3/1$	$3.14 \times 10^{-3}/3/1$
$a_2(50)$	$2.88 \times 10^{-2}/44/5$	$6.72 \times 10^{-2}/159/17$	$1.44 \times 10^{-1}/509/12$
$a_3(50)$	$2.57 \times 10^{-2}/63/8$	$7.24 \times 10^{-2}/237/25$	$6.79 \times 10^{-2}/259/7$
$a_4(50)$	$1.77 \times 10^{-2}/44/5$	$9.94 \times 10^{-2}/245/26$	$1.31 \times 10^{-1}/509/12$
$a_5(50)$	$1.74 \times 10^{-2}/44/5$	$6.06 \times 10^{-2}/159/17$	$1.66 \times 10^{-1}/509/12$
$a_6(50)$	$2.83 \times 10^{-3}/4/1$	$3.46 \times 10^{-3}/4/1$	$4.31 \times 10^{-3}/4/1$
$a_7(50)$	$1.88 \times 10^{-2}/44/5$	$5.47 \times 10^{-2}/159/17$	$1.40 \times 10^{-1}/509/12$
$a_8(50)$	$1.87 \times 10^{-2}/44/5$	$8.57 \times 10^{-2}/244/26$	$1.81 \times 10^{-1}/725/17$
$a_1(100)$	$3.06 \times 10^{-3}/3/1$	$2.70 \times 10^{-3}/3/1$	$4.55 \times 10^{-3}/3/1$
$a_2(100)$	$2.59 \times 10^{-2}/44/5$	$1.15 \times 10^{-1}/250/27$	$1.79 \times 10^{-1}/509/12$
$a_3(100)$	$3.02 \times 10^{-2}/63/8$	$8.74 \times 10^{-2}/197/21$	$8.90 \times 10^{-2}/259/7$
$a_4(100)$	$2.46 \times 10^{-2}/44/5$	$8.31 \times 10^{-2}/176/19$	$1.76 \times 10^{-1}/509/12$
$a_5(100)$	$2.40 \times 10^{-2}/44/5$	$1.17 \times 10^{-1}/250/27$	$1.83 \times 10^{-1}/509/12$
$a_6(100)$	$1.20 \times 10^{-2}/11/1$	$1.08 \times 10^{-2}/11/1$	$5.98 \times 10^{-3}/4/1$
$a_7(100)$	$2.57 \times 10^{-2}/44/5$	$1.21 \times 10^{-1}/250/27$	$1.97 \times 10^{-1}/509/12$
$a_8(100)$	$2.37 \times 10^{-2}/44/5$	$9.33 \times 10^{-2}/187/20$	$1.82 \times 10^{-1}/509/12$
$a_1(150)$	$3.87 \times 10^{-3}/3/1$	$1.92 \times 10^{-3}/3/1$	$6.30 \times 10^{-3}/3/1$
$a_2(150)$	$3.76 \times 10^{-2}/44/5$	$1.33 \times 10^{-1}/201/22$	$2.57 \times 10^{-1}/509/12$
$a_3(150)$	$4.78 \times 10^{-2}/63/8$	$1.15 \times 10^{-1}/179/19$	$1.22 \times 10^{-1}/259/7$
$a_4(150)$	$3.00 \times 10^{-2}/44/5$	$8.73 \times 10^{-2}/132/14$	$2.34 \times 10^{-1}/509/12$
$a_5(150)$	$3.16 \times 10^{-2}/44/5$	$1.34 \times 10^{-1}/201/22$	$2.51 \times 10^{-1}/509/12$
$a_6(150)$	$1.60 \times 10^{-2}/11/1$	$1.42 \times 10^{-2}/11/1$	$1.33 \times 10^{-2}/8/1$
$a_7(150)$	$3.09 \times 10^{-2}/44/5$	$1.32 \times 10^{-1}/201/22$	$2.46 \times 10^{-1}/509/12$
$a_8(150)$	$3.17 \times 10^{-2}/44/5$	$2.09 \times 10^{-1}/301/32$	$3.02 \times 10^{-1}/642/24$

Table 5. Numerical results for Problem 5.

Inti(n)	ILR	VRMILP	DFPRPMHS
	CPUT/Nfunc/Niter	CPUT/Nfunc/Niter	CPUT/Nfunc/Niter
$a_1(5)$	$5.45 \times 10^{-2}/283/36$	$3.93 \times 10^{-2}/309/34$	$6.74 \times 10^{-2}/694/43$
$a_2(5)$	$2.50 \times 10^{-2}/235/30$	$3.07 \times 10^{-2}/330/36$	$9.82 \times 10^{-2}/1113/46$
$a_3(5)$	$2.09 \times 10^{-2}/191/24$	$2.60 \times 10^{-2}/294/32$	$5.82 \times 10^{-2}/663/50$
$a_4(5)$	$2.41 \times 10^{-2}/235/30$	$3.07 \times 10^{-2}/330/36$	$8.84 \times 10^{-2}/1125/46$
$a_5(5)$	$2.04 \times 10^{-2}/212/27$	$3.62 \times 10^{-2}/289/31$	$1.26 \times 10^{-1}/1142/56$
$a_6(5)$	$2.66 \times 10^{-2}/253/32$	$2.25 \times 10^{-2}/286/32$	$1.41 \times 10^{-1}/1265/113$
$a_7(5)$	$2.84 \times 10^{-2}/212/27$	$3.94 \times 10^{-2}/289/31$	$1.22 \times 10^{-1}/1142/56$
$a_8(5)$	$3.33 \times 10^{-2}/247/32$	$3.92 \times 10^{-2}/331/35$	$2.61 \times 10^{-1}/2555/55$
$a_1(10)$	$6.23 \times 10^{-2}/254/32$	$9.28 \times 10^{-2}/356/39$	$1.72 \times 10^{-1}/757/57$
$a_2(10)$	$5.84 \times 10^{-2}/239/31$	$6.88 \times 10^{-2}/308/33$	$1.68 \times 10^{-1}/800/48$
$a_3(10)$	$5.94 \times 10^{-2}/258/33$	$6.19 \times 10^{-2}/296/33$	$1.20 \times 10^{-1}/565/46$
$a_4(10)$	$5.24 \times 10^{-2}/232/30$	$6.53 \times 10^{-2}/308/33$	$1.60 \times 10^{-1}/800/48$
$a_5(10)$	$5.56 \times 10^{-2}/239/31$	$6.79 \times 10^{-2}/325/35$	$1.65 \times 10^{-1}/789/62$
$a_6(10)$	$4.87 \times 10^{-2}/205/26$	$6.13 \times 10^{-2}/282/32$	$1.81 \times 10^{-1}/887/53$
$a_7(10)$	$5.46 \times 10^{-2}/239/31$	$6.68 \times 10^{-2}/325/35$	$1.63 \times 10^{-1}/789/62$
$a_8(10)$	$6.69 \times 10^{-2}/275/36$	$7.43 \times 10^{-2}/350/37$	$2.56 \times 10^{-1}/1297/54$
$a_1(50)$	$2.56 \times 10^{-1}/211/27$	$3.13 \times 10^{-1}/308/33$	$5.78 \times 10^{-1}/657/46$
$a_2(50)$	$2.42 \times 10^{-1}/229/29$	$2.56 \times 10^{-1}/279/31$	$1.24 \times 10^0/1319/111$
$a_3(50)$	$2.35 \times 10^{-1}/250/32$	$2.68 \times 10^{-1}/308/33$	$6.76 \times 10^{-1}/682/59$
$a_4(50)$	$2.38 \times 10^{-1}/229/29$	$2.83 \times 10^{-1}/279/31$	$1.17 \times 10^0/1276/105$
$a_5(50)$	$2.33 \times 10^{-1}/228/29$	$3.54 \times 10^{-1}/340/38$	$9.50 \times 10^{-1}/983/83$
$a_6(50)$	$2.17 \times 10^{-1}/219/28$	$2.92 \times 10^{-1}/324/35$	$8.49 \times 10^{-1}/937/52$
$a_7(50)$	$2.46 \times 10^{-1}/228/29$	$3.07 \times 10^{-1}/340/38$	$9.22 \times 10^{-1}/983/83$
$a_8(50)$	$2.32 \times 10^{-1}/241/31$	$3.16 \times 10^{-1}/364/39$	$1.09 \times 10^0/1238/56$
$a_1(100)$	$3.79 \times 10^{-1}/220/28$	$4.93 \times 10^{-1}/325/35$	$1.12 \times 10^0/753/44$
$a_2(100)$	$4.04 \times 10^{-1}/236/30$	$4.62 \times 10^{-1}/283/32$	$1.24 \times 10^0/768/61$
$a_3(100)$	$4.28 \times 10^{-1}/250/32$	$4.92 \times 10^{-1}/317/34$	$1.87 \times 10^0/1229/56$
$a_4(100)$	$3.82 \times 10^{-1}/236/30$	$4.27 \times 10^{-1}/273/31$	$1.24 \times 10^0/768/61$
$a_5(100)$	$4.77 \times 10^{-1}/262/33$	$5.63 \times 10^{-1}/331/36$	$1.40 \times 10^0/892/65$
$a_6(100)$	$4.27 \times 10^{-1}/254/33$	$5.55 \times 10^{-1}/314/34$	$1.66 \times 10^0/1020/50$
$a_7(100)$	$4.73 \times 10^{-1}/262/33$	$5.60 \times 10^{-1}/331/36$	$1.30 \times 10^0/892/65$
$a_8(100)$	$4.79 \times 10^{-1}/291/38$	$5.78 \times 10^{-1}/380/41$	$1.96 \times 10^0/1246/58$
$a_1(150)$	$5.67 \times 10^{-1}/221/28$	$6.78 \times 10^{-1}/287/31$	$2.40 \times 10^0/1010/50$
$a_2(150)$	$5.10 \times 10^{-1}/206/26$	$8.62 \times 10^{-1}/375/41$	$2.98 \times 10^0/1327/123$
$a_3(150)$	$5.57 \times 10^{-1}/228/29$	$6.93 \times 10^{-1}/325/35$	$1.95 \times 10^0/832/52$
$a_4(150)$	$5.48 \times 10^{-1}/206/26$	$8.80 \times 10^{-1}/375/41$	$3.41 \times 10^0/1363/126$
$a_5(150)$	$5.66 \times 10^{-1}/214/27$	$8.50 \times 10^{-1}/358/40$	$2.31 \times 10^0/1031/78$
$a_6(150)$	$4.97 \times 10^{-1}/240/31$	$7.32 \times 10^{-1}/313/34$	$1.83 \times 10^0/875/46$
$a_7(150)$	$5.95 \times 10^{-1}/214/27$	$9.02 \times 10^{-1}/358/40$	$2.42 \times 10^0/1031/78$
$a_8(150)$	$6.91 \times 10^{-1}/291/38$	$1.13 \times 10^0/456/49$	$2.85 \times 10^0/1242/58$

Table 6. Numerical results for Problem 6.

Inti(n)	ILR	VRMILP	DFPRPMHS
	CPUT/Nfunc/Niter	CPUT/Nfunc/Niter	CPUT/Nfunc/Niter
$a_1(5)$	$7.08 \times 10^{-3}/\mathbf{34/14}$	$\mathbf{6.92} \times 10^{-3}/84/22$	$4.32 \times 10^{-2}/672/21$
$a_2(5)$	$7.84 \times 10^{-3}/\mathbf{69/27}$	$\mathbf{5.48} \times 10^{-3}/71/22$	$2.96 \times 10^{-2}/513/40$
$a_3(5)$	$\mathbf{4.14} \times 10^{-3}/\mathbf{34/14}$	$7.29 \times 10^{-3}/111/27$	$2.58 \times 10^{-2}/504/34$
$a_4(5)$	$7.08 \times 10^{-3}/74/28$	$\mathbf{4.61} \times 10^{-3}/\mathbf{71/22}$	$2.23 \times 10^{-2}/414/41$
$a_5(5)$	$5.92 \times 10^{-3}/\mathbf{69/27}$	$\mathbf{4.87} \times 10^{-3}/71/22$	$2.85 \times 10^{-2}/513/40$
$a_6(5)$	$3.27 \times 10^{-3}/32/13$	$\mathbf{2.77} \times 10^{-3}/\mathbf{28/11}$	$3.18 \times 10^{-2}/670/20$
$a_7(5)$	$6.28 \times 10^{-3}/\mathbf{69/27}$	$\mathbf{4.86} \times 10^{-3}/71/22$	$2.74 \times 10^{-2}/513/40$
$a_8(5)$	$6.68 \times 10^{-3}/\mathbf{79/30}$	$\mathbf{5.76} \times 10^{-3}/84/25$	$2.66 \times 10^{-2}/446/40$
$a_1(10)$	$\mathbf{6.94} \times 10^{-3}/\mathbf{31/13}$	$8.81 \times 10^{-3}/67/20$	$4.26 \times 10^{-2}/424/29$
$a_2(10)$	$\mathbf{6.88} \times 10^{-3}/\mathbf{34/14}$	$1.25 \times 10^{-2}/95/27$	$6.07 \times 10^{-2}/672/21$
$a_3(10)$	$\mathbf{6.52} \times 10^{-3}/\mathbf{31/13}$	$1.68 \times 10^{-2}/142/31$	$3.96 \times 10^{-2}/384/30$
$a_4(10)$	$\mathbf{6.86} \times 10^{-3}/\mathbf{34/14}$	$1.36 \times 10^{-2}/90/26$	$6.88 \times 10^{-2}/672/21$
$a_5(10)$	$\mathbf{7.18} \times 10^{-3}/\mathbf{34/14}$	$1.23 \times 10^{-2}/95/27$	$6.25 \times 10^{-2}/672/21$
$a_6(10)$	$7.48 \times 10^{-3}/35/14$	$\mathbf{4.64} \times 10^{-3}/\mathbf{33/13}$	$5.18 \times 10^{-2}/561/18$
$a_7(10)$	$\mathbf{6.86} \times 10^{-3}/\mathbf{34/14}$	$1.20 \times 10^{-2}/95/27$	$6.47 \times 10^{-2}/672/21$
$a_8(10)$	$\mathbf{6.92} \times 10^{-3}/\mathbf{34/14}$	$1.10 \times 10^{-2}/86/25$	$6.35 \times 10^{-2}/672/21$
$a_1(50)$	$\mathbf{3.20} \times 10^{-2}/\mathbf{41/17}$	$3.59 \times 10^{-1}/1099/94$	$2.04 \times 10^{-1}/698/31$
$a_2(50)$	$\mathbf{2.57} \times 10^{-2}/\mathbf{36/15}$	$4.34 \times 10^{-2}/105/29$	$1.82 \times 10^{-1}/572/23$
$a_3(50)$	$\mathbf{2.34} \times 10^{-2}/\mathbf{38/16}$	$2.38 \times 10^{-1}/769/56$	$1.70 \times 10^{-1}/511/38$
$a_4(50)$	$\mathbf{2.23} \times 10^{-2}/\mathbf{36/15}$	$3.88 \times 10^{-2}/101/28$	$1.87 \times 10^{-1}/572/23$
$a_5(50)$	$\mathbf{2.68} \times 10^{-2}/\mathbf{36/15}$	$4.40 \times 10^{-2}/105/29$	$1.84 \times 10^{-1}/572/23$
$a_6(50)$	$2.29 \times 10^{-2}/37/15$	$\mathbf{1.08} \times 10^{-2}/\mathbf{25/10}$	$2.13 \times 10^{-1}/672/21$
$a_7(50)$	$\mathbf{2.70} \times 10^{-2}/\mathbf{36/15}$	$5.36 \times 10^{-2}/105/29$	$1.88 \times 10^{-1}/572/23$
$a_8(50)$	$\mathbf{2.61} \times 10^{-2}/\mathbf{36/15}$	$4.65 \times 10^{-2}/102/28$	$1.88 \times 10^{-1}/572/23$
$a_1(100)$	$\mathbf{3.96} \times 10^{-2}/\mathbf{41/17}$	$5.61 \times 10^{-2}/84/25$	$2.28 \times 10^{-1}/569/22$
$a_2(100)$	$\mathbf{5.49} \times 10^{-2}/\mathbf{57/24}$	$6.17 \times 10^{-2}/100/28$	$2.95 \times 10^{-1}/676/23$
$a_3(100)$	$\mathbf{3.64} \times 10^{-2}/\mathbf{41/17}$	$4.73 \times 10^{-2}/84/23$	$2.08 \times 10^{-1}/411/43$
$a_4(100)$	$\mathbf{4.85} \times 10^{-2}/\mathbf{57/24}$	$6.49 \times 10^{-2}/100/28$	$2.92 \times 10^{-1}/676/23$
$a_5(100)$	$\mathbf{4.61} \times 10^{-2}/\mathbf{57/24}$	$5.92 \times 10^{-2}/100/28$	$2.79 \times 10^{-1}/676/23$
$a_6(100)$	$3.35 \times 10^{-2}/39/16$	$\mathbf{2.28} \times 10^{-2}/\mathbf{33/13}$	$2.64 \times 10^{-1}/673/22$
$a_7(100)$	$\mathbf{4.73} \times 10^{-2}/\mathbf{57/24}$	$6.31 \times 10^{-2}/100/28$	$2.89 \times 10^{-1}/676/23$
$a_8(100)$	$7.09 \times 10^{-2}/\mathbf{83/33}$	$\mathbf{5.68} \times 10^{-2}/100/28$	$3.00 \times 10^{-1}/676/23$
$a_1(150)$	$\mathbf{4.56} \times 10^{-2}/\mathbf{38/16}$	$6.69 \times 10^{-2}/83/24$	$4.36 \times 10^{-1}/678/24$
$a_2(150)$	$\mathbf{5.06} \times 10^{-2}/\mathbf{41/17}$	$1.01 \times 10^{-1}/125/35$	$3.39 \times 10^{-1}/512/39$
$a_3(150)$	$\mathbf{4.98} \times 10^{-2}/\mathbf{38/16}$	$1.21 \times 10^{-1}/154/34$	$4.10 \times 10^{-1}/661/52$
$a_4(150)$	$\mathbf{4.84} \times 10^{-2}/\mathbf{41/17}$	$1.21 \times 10^{-1}/125/35$	$3.36 \times 10^{-1}/517/41$
$a_5(150)$	$\mathbf{4.28} \times 10^{-2}/\mathbf{41/17}$	$1.23 \times 10^{-1}/125/35$	$3.83 \times 10^{-1}/512/39$
$a_6(150)$	$6.04 \times 10^{-2}/39/16$	$\mathbf{3.64} \times 10^{-2}/\mathbf{35/14}$	$3.85 \times 10^{-1}/674/22$
$a_7(150)$	$\mathbf{4.13} \times 10^{-2}/\mathbf{41/17}$	$9.58 \times 10^{-2}/125/35$	$3.00 \times 10^{-1}/512/39$
$a_8(150)$	$\mathbf{4.37} \times 10^{-2}/\mathbf{41/17}$	$1.17 \times 10^{-1}/129/36$	$3.59 \times 10^{-1}/597/33$

Table 7. Numerical results for Problem 7.

Inti(n)	ILR	VRMILP	DFPRPMHS
	CPUT/Nfunc/Niter	CPUT/Nfunc/Niter	CPUT/Nfunc/Niter
$a_1(5)$	$1.75 \times 10^{-2}/111/18$	$6.37 \times 10^{-3}/73/12$	$4.54 \times 10^{-2}/752/32$
$a_2(5)$	$9.71 \times 10^{-3}/118/19$	$1.35 \times 10^{-2}/207/29$	$3.64 \times 10^{-2}/811/55$
$a_3(5)$	$5.77 \times 10^{-3}/104/17$	$3.63 \times 10^{-3}/73/12$	$3.20 \times 10^{-2}/829/17$
$a_4(5)$	$6.72 \times 10^{-3}/126/20$	$5.25 \times 10^{-3}/112/17$	$3.93 \times 10^{-2}/934/46$
$a_5(5)$	$6.59 \times 10^{-3}/118/19$	$9.84 \times 10^{-3}/207/29$	$3.90 \times 10^{-2}/891/51$
$a_6(5)$	$5.47 \times 10^{-3}/96/17$	$3.57 \times 10^{-3}/72/13$	$4.04 \times 10^{-2}/1024/21$
$a_7(5)$	$6.59 \times 10^{-3}/118/19$	$1.00 \times 10^{-2}/207/29$	$3.93 \times 10^{-2}/892/51$
$a_8(5)$	$6.74 \times 10^{-3}/128/20$	$9.35 \times 10^{-3}/207/29$	$4.15 \times 10^{-2}/927/45$
$a_1(10)$	$1.70 \times 10^{-2}/113/18$	$8.21 \times 10^{-3}/73/12$	$6.53 \times 10^{-2}/735/43$
$a_2(10)$	$1.72 \times 10^{-2}/148/22$	$1.30 \times 10^{-2}/137/20$	$8.32 \times 10^{-2}/1047/35$
$a_3(10)$	$1.36 \times 10^{-2}/133/20$	$6.79 \times 10^{-3}/73/12$	$6.49 \times 10^{-2}/830/17$
$a_4(10)$	$1.71 \times 10^{-2}/138/22$	$1.12 \times 10^{-2}/112/17$	$7.33 \times 10^{-2}/913/30$
$a_5(10)$	$1.73 \times 10^{-2}/148/22$	$1.28 \times 10^{-2}/137/20$	$1.07 \times 10^{-1}/1047/35$
$a_6(10)$	$1.19 \times 10^{-2}/96/17$	$8.66 \times 10^{-3}/72/13$	$8.01 \times 10^{-2}/1024/21$
$a_7(10)$	$1.66 \times 10^{-2}/148/22$	$1.44 \times 10^{-2}/137/20$	$8.95 \times 10^{-2}/1047/35$
$a_8(10)$	$1.54 \times 10^{-2}/131/21$	$2.27 \times 10^{-2}/241/33$	$7.95 \times 10^{-2}/914/43$
$a_1(50)$	$4.83 \times 10^{-2}/118/19$	$2.48 \times 10^{-2}/79/13$	$2.08 \times 10^{-1}/836/16$
$a_2(50)$	$4.34 \times 10^{-2}/133/21$	$4.56 \times 10^{-2}/185/26$	$2.22 \times 10^{-1}/988/27$
$a_3(50)$	$3.22 \times 10^{-2}/105/17$	$2.33 \times 10^{-2}/79/13$	$1.81 \times 10^{-1}/772/37$
$a_4(50)$	$3.98 \times 10^{-2}/133/21$	$2.78 \times 10^{-2}/112/17$	$2.34 \times 10^{-1}/999/52$
$a_5(50)$	$4.44 \times 10^{-2}/133/21$	$4.96 \times 10^{-2}/185/26$	$2.44 \times 10^{-1}/988/27$
$a_6(50)$	$3.19 \times 10^{-2}/102/18$	$2.16 \times 10^{-2}/78/14$	$2.29 \times 10^{-1}/1024/21$
$a_7(50)$	$4.29 \times 10^{-2}/133/21$	$4.79 \times 10^{-2}/185/26$	$2.28 \times 10^{-1}/988/27$
$a_8(50)$	$4.32 \times 10^{-2}/133/21$	$5.52 \times 10^{-2}/209/29$	$2.13 \times 10^{-1}/931/27$
$a_1(100)$	$6.02 \times 10^{-2}/118/19$	$3.75 \times 10^{-2}/79/13$	$2.51 \times 10^{-1}/838/16$
$a_2(100)$	$5.16 \times 10^{-2}/133/21$	$6.32 \times 10^{-2}/205/29$	$2.91 \times 10^{-1}/1018/31$
$a_3(100)$	$4.38 \times 10^{-2}/117/19$	$2.48 \times 10^{-2}/79/13$	$2.37 \times 10^{-1}/849/18$
$a_4(100)$	$5.98 \times 10^{-2}/133/21$	$4.46 \times 10^{-2}/112/17$	$2.75 \times 10^{-1}/1018/31$
$a_5(100)$	$5.15 \times 10^{-2}/133/21$	$6.67 \times 10^{-2}/205/29$	$2.78 \times 10^{-1}/1018/31$
$a_6(100)$	$4.26 \times 10^{-2}/102/18$	$2.68 \times 10^{-2}/78/14$	$2.74 \times 10^{-1}/1024/21$
$a_7(100)$	$5.38 \times 10^{-2}/133/21$	$8.67 \times 10^{-2}/205/29$	$3.08 \times 10^{-1}/1018/31$
$a_8(100)$	$5.41 \times 10^{-2}/130/21$	$9.41 \times 10^{-2}/241/33$	$3.10 \times 10^{-1}/1058/36$
$a_1(150)$	$5.88 \times 10^{-2}/111/18$	$4.44 \times 10^{-2}/79/13$	$3.23 \times 10^{-1}/838/16$
$a_2(150)$	$6.85 \times 10^{-2}/139/22$	$7.94 \times 10^{-2}/168/24$	$4.13 \times 10^{-1}/1048/35$
$a_3(150)$	$7.80 \times 10^{-2}/142/22$	$3.71 \times 10^{-2}/79/13$	$3.38 \times 10^{-1}/787/49$
$a_4(150)$	$6.78 \times 10^{-2}/132/21$	$5.61 \times 10^{-2}/127/19$	$3.81 \times 10^{-1}/1018/31$
$a_5(150)$	$7.81 \times 10^{-2}/139/22$	$8.21 \times 10^{-2}/168/24$	$4.24 \times 10^{-1}/1048/35$
$a_6(150)$	$5.85 \times 10^{-2}/108/19$	$4.74 \times 10^{-2}/78/14$	$4.21 \times 10^{-1}/1024/21$
$a_7(150)$	$7.66 \times 10^{-2}/139/22$	$8.18 \times 10^{-2}/168/24$	$3.88 \times 10^{-1}/1048/35$
$a_8(150)$	$8.15 \times 10^{-2}/149/24$	$5.84 \times 10^{-2}/118/18$	$4.22 \times 10^{-1}/1056/36$

To provide a clearer characterization of the performance differences among the three algorithms, we adopt the performance profiles proposed by Dolan and Moré [28]. These profiles evaluate algorithmic behavior based on several key performance indicators, specifically the CPU time in seconds, the number of function evaluations, and the number of iterations. By plotting these indicators, the profiles offer a visual and comparative summary of algorithm efficiency. In these plots, a higher performance curve corresponds to better overall performance, making interpretation both intuitive and informative. By drawing these performance profiles for these three algorithms, we can visually assess and compare their efficiency, as shown in Figures 1–3. According to Figure 1, the ILR algorithm demonstrates significant efficiency, solving approximately 56% of the test problems with the lowest CPUT compared to the VRMILP and DFPRPMHS algorithms, which solve around 44% and 4% of the test problems, respectively. Similarly, Figure 2 shows that the ILR algorithm maintains its superior performance, solving approximately 75% of the test problems with the fewest Nfunc. In contrast, the VRMILP and DFPRPMHS algorithms solve about 26% and 7% of the test problems, respectively, with the least number of function evaluations. Lastly, Figure 3 further confirms the ILR algorithm's efficiency, solving approximately 53% of the test problems with the fewest Niter, while the VRMILP and DFPRPMHS algorithms solve around 37% and 20% of the test problems, respectively, with the fewest iterations.

Overall, these performance profiles highlight the ILR algorithm's effectiveness in solving large-scale nonlinear systems of equations with convex constraints, outperforming the VRMILP and DFPRPMHS algorithms across multiple performance metrics.

5. Numerical experiments for impulse noise image restoration

In this section, we extend the evaluation of the proposed ILR algorithm to impulse noise image restoration problems. To validate the effectiveness of the ILR algorithm, we apply it to benchmark grayscale images subjected to varying levels of impulse noise.

5.1. General setups

Impulse noise image restoration is a critical topic in the field of image processing, particularly due to its importance in improving the quality of images corrupted by noise. Noise in images can be introduced through various sources, such as malfunctioning pixels in camera sensors, faulty memory locations in hardware, or transmission errors in communication channels. Common types of noise include Gaussian noise and impulse noise, with the latter often manifesting as salt-and-pepper noise. To address the challenge of removing impulse noise, Chan et al. [29] proposed a two-phase denoising scheme. This scheme combines the adaptive median filter (AMF) method with a variational method to effectively detect and restore noisy pixels.

Let $m \times n$ denote the pixel size of an original image. The pixel locations are indexed by the set $\mathcal{M} = \{1, 2, \dots, m\} \times \{1, 2, \dots, n\}$. We denote the noise candidate set by $\mathcal{N} \subset \mathcal{M}$, and $|\mathcal{N}|$ represents the number of elements in \mathcal{N} . In the first phase, noise detection is performed using an AMF. For a pixel located at $(i, j) \in \mathcal{M}$, the observed pixel value is denoted by y_{ij} , and the neighborhood of pixel (i, j) is defined as $\mathcal{V}_{ij} = \{(i, j - 1), (i, j + 1), (i - 1, j), (i + 1, j)\}$. The AMF detects noise by considering these neighborhood values. Once the noisy pixels are detected, the second phase involves the restoration of

these pixels. This is achieved by minimizing the following regularization function:

$$\min_{\mathbf{x}} \sum_{(i,j) \in \mathcal{N}} \left[|x_{i,j} - y_{i,j}| + \frac{\beta}{2} (2\Phi_{i,j}^1 + \Phi_{i,j}^2) \right],$$

where

$$\Phi_{i,j}^1 = \sum_{(m,n) \in \mathcal{V}_{ij} \setminus \mathcal{N}} \varphi_{\alpha}(x_{i,j} - y_{m,n}), \quad \Phi_{i,j}^2 = \sum_{(m,n) \in \mathcal{V}_{ij} \setminus \mathcal{N}} \varphi_{\alpha}(x_{i,j} - u_{m,n}).$$

Here, β is a regularization parameter, and $\varphi_{\alpha}(\cdot)$ is an even edge-preserving potential function with parameter $\alpha > 0$. The vector $\mathbf{x} = [x_{i,j}]_{(i,j) \in \mathcal{N}}$ is optimized lexicographically to achieve denoising. The regularization problem posed in the second phase is nonsmooth due to the data-fitting term $|x_{i,j} - y_{i,j}|$. To address this, Cai et al. [30] proposed removing the nonsmooth term and instead solving the following smooth unconstrained optimization problem:

$$\min_{\mathbf{x}} f_{\alpha}(\mathbf{x}) := \sum_{(i,j) \in \mathcal{N}} (2\Phi_{i,j}^1 + \Phi_{i,j}^2).$$

The potential function $\varphi_{\alpha}(\cdot)$ plays a crucial role in preserving edges while smoothing the image. A commonly used potential function is the Huber function, defined as:

$$\varphi_{\alpha}(t) = \begin{cases} \frac{t^2}{2\alpha}, & \text{for } |t| \leq \alpha, \\ |t| - \frac{\alpha}{2}, & \text{for } |t| > \alpha. \end{cases}$$

This function is convex and first-order Lipschitz continuous, making it suitable for the minimization problems described above. Let $\nabla f_{\alpha}(\mathbf{x})$ denote the gradient of the function $f_{\alpha}(\mathbf{x})$. In alignment with Proposition 6 in [30], if φ_{α} is convex, then $\nabla f_{\alpha}(\mathbf{x})$ is monotone.

5.2. Numerical reports

In this section, all parameters for these three algorithms are set as described in Section 4. The stopping criteria for these three algorithms are defined as follows:

$$\frac{\|\mathbf{x}_k - \mathbf{x}_{k-1}\|}{\|\mathbf{x}_k\|} \leq \tau \quad \text{or} \quad \frac{|f_{\alpha}(\mathbf{x}_k) - f_{\alpha}(\mathbf{x}_{k-1})|}{|f_{\alpha}(\mathbf{x}_k)|} \leq \tau.$$

For this experiment, we utilize the well-known grayscale test images Man (1024×1024) and Tank2 (512×512), which are sourced from the website <https://www.hlevkin.com>. We examine the performance of these three algorithms by applying them to images corrupted with 30% and 70% impulse noise. The noisy images, as well as the images recovered by these three algorithms, are shown in Figures 4 and 5. The corresponding numerical results are provided in Table 8. Based on these figures and the table, we can draw the following conclusions: (1) Figures 4 and 5 demonstrate that all three algorithms successfully recover the images affected by 30% and 70% impulse noise; (2) Recovering an image with 30% impulse noise requires less CPU time and fewer iterations compared to recovering an image with 70% impulse noise; (3) Among these three algorithms, the ILR algorithm generally requires less CPU time and fewer iterations than the VRMILP and DFPRPMHS algorithms for a given level of impulse noise.



Figure 4. From left to right: A noisy image with 30% impulse noise and recovered images obtained by the ILR, VRMILP, and DFPRPMHS algorithms.



Figure 5. From left to right: A noisy image with 70% impulse noise and recovered images obtained by the ILR, VRMILP, and DFPRPMHS algorithms.

Table 8. The numerical results of the ILR, VRMILP, and DFPRPMHS algorithms.

Algorithm	Man		Tank	
	Noise: 30%	Noise: 70%	Noise: 30%	Noise: 70%
	Niter/CPUT	Niter/CPUT	Niter/CPUT	Niter/CPUT
ILR	14/13.38	29/29.04	8/1.46	16/3.66
VRMILP	22/28.26	39/55.89	18/5.47	20/6.52
DFPRPMHS	27/38.08	42/46.95	17/2.59	18/3.92

6. Conclusions

In this paper, we presented an improved LS-RMIL-type conjugate gradient projection algorithm aimed at efficiently solving systems of nonlinear equations with convex constraints. The proposed algorithm demonstrates several key advantages, including the ability to generate search directions that satisfy sufficient descent and trust-region properties independently of the line search approach. Additionally, the proposed algorithm only requires continuous and monotone assumptions for systems of nonlinear equations, which makes it applicable under less restrictive conditions compared to existing methods. We established the global convergence of the proposed algorithm without relying on the Lipschitz continuity assumption, further relaxing the conditions that need to be satisfied for successful implementation. Extensive numerical simulations, including large-scale systems of nonlinear equations and impulse noise image restoration problems, have shown that the proposed algorithm exhibits superior efficiency and stability compared to existing algorithms. These results indicate that the proposed algorithm is a promising and competitive approach, with significant potential for practical applications, such as image restoration.

Author contributions

Yan Xia: Conceptualization, Investigation, Writing—original draft, Writing—review and editing, Funding acquisition; Xuejie Ma: Writing—review and editing, Funding acquisition; Dandan Li: Conceptualization, Funding acquisition, Writing—review and editing. All authors have read and approved the final version of the manuscript for publication.

Use of Generative-AI tools declaration

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

Acknowledgments

This work is supported by the Guangzhou Huashang College Daoshi Project (2024HSDS28).

Conflict of interest

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

References

1. H. Chen, Y. Wang, H. Zhao, Finite convergence of a projected proximal point algorithm for the generalized variational inequalities, *Oper. Res. Lett.*, **40** (2012), 303–305. <https://doi.org/10.1016/j.orl.2012.03.011>
2. Y. Wang, L. Qi, S. Luo, An alternative steepest direction method for the optimization in evaluating geometric discord, *Pacific J. Optim.*, **10** (2014), 137–149.
3. N. Iusem, V. Solodov, Newton-type methods with generalized distances for constrained optimization, *Optimization*, **41** (1997), 257–278. <https://doi.org/10.1080/02331939708844339>
4. P. Liu, L. Li, H. Shao, M. Liu, J. Fan, An inertial-type CG projection method with restart for pseudo-monotone costs with application to traffic assignment, *Netw. Spat. Econ.*, **25** (2024), 147–172. <https://doi.org/10.1007/s11067-024-09653-z>
5. D. Li, J. Wu, Y. Li, S. Wang, A modified spectral gradient projection-based algorithm for large-scale constrained nonlinear equations with applications in compressive sensing, *J. Comput. Appl. Math.*, **424** (2023), 115006. <https://doi.org/10.1016/j.cam.2022.115006>
6. Y. Zhao, D. Li, Monotonicity of fixed point and normal mappings associated with variational inequality and its application, *SIAM J. Optim.*, **11** (2001), 962–973. <https://doi.org/10.1137/S1052623499357957>
7. D. Li, S. Wang, Y. Li, J. Wu, A projection-based hybrid PRP-DY type conjugate gradient algorithm for constrained nonlinear equations with applications, *Appl. Numer. Math.*, **195** (2024), 105–125. <https://doi.org/10.1016/j.apnum.2023.09.009>
8. D. Li, Y. Li, S. Wang, An improved three-term conjugate gradient algorithm for constrained nonlinear equations under non-lipschitz conditions and its Applications, *Mathematics*, **12** (2024), 2556. <https://doi.org/10.3390/math12162556>
9. Z. Liu, X. Zhang, M. Su, Convergence analysis of Newton-Raphson method in feasible power-flow for DC network, *IEEE Trans. Power Syst.*, **35** (2020), 4100–4103. <https://doi.org/10.1109/TPWRS.2020.2986706>
10. N. Barnafi, L. Pavarino, S. Scacchi, Parallel inexact Newton-Krylov and quasi-Newton solvers for nonlinear elasticity, *Comput. Methods Appl. Mech. Eng.*, **400** (2022) 115557. <https://doi.org/10.1016/j.cma.2022.115557>
11. A. Mahdavi, S. Salehi, A superlinearly convergent nonmonotone quasi-Newton method for unconstrained multiobjective optimization, *Optim. Methods Software*, **35** (2020) 1223–1247. <https://doi.org/10.1080/10556788.2020.1737691>
12. R. Sihwail, O. Solaiman, K. Omar, A hybrid approach for solving systems of nonlinear equations using harris hawks optimization and newtons method, *IEEE Access*, **9** (2021), 95791–95807. <https://doi.org/10.1109/ACCESS.2021.3094471>
13. V. Krutikov, E. Tovbis, P. Stanimirovic, On the convergence rate of quasi-Newton methods on strongly convex functions with lipschitz gradient, *Mathematics*, **11** (2023), 4715. <https://doi.org/10.3390/math11234715>

14. G. Yuan, T. Li, W. Hu, A conjugate gradient algorithm for large-scale nonlinear equations and image restoration problems, *Appl. Numer. Math.*, **147** (2020), 129–141. <https://doi.org/10.1016/j.apnum.2019.08.022>
15. M. Koorapetse, P. Kaelo, An efficient hybrid conjugate gradient-based projection method for convex constrained nonlinear monotone equations, *J. Interdiscip. Math.*, **22** (2019), 1031–1050. <https://doi.org/10.1080/09720502.2019.1700889>
16. K. Danmalam, H. Mohammad, A. Abubakar, Hybrid algorithm for system of nonlinear monotone equations based on the convex combination of Fletcher-Reeves and a new conjugate residual parameters, *Thai J. Math.*, **18** (2020), 2093–2106.
17. M. Waziri, A. Yusuf, A. Abubakar, Improved conjugate gradient method for nonlinear system of equations, *Comput. Appl. Math.*, **39** (2020), 321. <https://doi.org/10.1007/s40314-020-01374-6>
18. T. Song, Z. Liu, An efficient subspace minimization conjugate gradient method for solving nonlinear monotone equations with convex constraints, *Axioms*, **13** (2024), 170. <https://doi.org/10.3390/axioms13030170>
19. G. Ma, J. Jin, J. Jian, A modified inertial three-term conjugate gradient projection method for constrained nonlinear equations with applications in compressed sensing, *Numer. Algor.*, **92** (2023), 1621–1653. <https://doi.org/10.1007/s11075-022-01356-1>
20. W. Liu, J. Jian, J. Yin, An inertial spectral conjugate gradient projection method for constrained nonlinear pseudo-monotone equations, *Numer. Algor.*, **97** (2024), 985–1015. <https://doi.org/10.1007/s11075-023-01736-1>
21. J. Sabiu, A. Shah, P. Stanimirovic, Modified optimal Perry conjugate gradient method for solving system of monotone equations with applications, *Appl. Numer. Math.*, **184** (2023), 431–445. <https://doi.org/10.1016/j.apnum.2022.10.016>
22. Y. Liu, C. Storey, Efficient generalized conjugate gradient algorithms, part 1: Theory, *J. Optim. Theory Appl.*, **69** (1991), 129–137. <https://doi.org/10.1007/BF00940464>
23. M. Rivaie, M. Mamat, L. June, A new class of nonlinear conjugate gradient coefficients with global convergence properties, *Appl. Math. Comput.*, **218** (2012), 11323–11332. <https://doi.org/10.1016/j.amc.2012.05.030>
24. P. Liu, H. Shao, Z. Yuan, X. Wu, T. Zheng, A family of three-term conjugate gradient projection methods with a restart procedure and their relaxed-inertial extensions for the constrained nonlinear pseudo-monotone equations with applications, *Numer. Algor.*, **94** (2023), 1055–1083. <https://doi.org/10.1007/s11075-023-01527-8>
25. P. Liu, X. Wu, H. Shao, Y. Zhang, S. Cao, Three adaptive hybrid derivative-free projection methods for constrained monotone nonlinear equations and their applications, *Numer. Linear Algebra Appl.*, **30** (2023), e2471. <https://doi.org/10.1002/nla.2471>
26. X. Wu, H. Shao, P. Liu, An efficient conjugate gradient-based algorithm for unconstrained optimization and its projection extension to large-scale constrained nonlinear equations with applications in signal recovery and image denoising problems, *J. Comput. Appl. Math.*, **422** (2023), 114879. <https://doi.org/10.1016/j.cam.2022.114879>

27. A. Ibrahim, P. Kumam, W. Kumam, A family of derivative-free conjugate gradient methods for constrained nonlinear equations and image restoration, *IEEE Access*, **8** (2020), 162714–162729. <https://doi.org/10.1109/ACCESS.2020.3020969>
28. E. Dolan, J. Jorge, Benchmarking optimization software with performance profiles, *Math. Program.*, **91** (2001), 201–213. <https://doi.org/10.1007/s101070100263>
29. R. Chan, C. Ho, M. Nikolova, Salt-and-pepper noise removal by median-type noise detectors and detail-preserving regularization, *IEEE Trans. Image Process.*, **14** (2005), 1479–1485. <https://doi.org/10.1109/TIP.2005.852196>
30. J. Cai, R. Chan, D. Fiore, Minimization of a detail-preserving regularization functional for impulse noise removal, *J. Math. Imaging Vis.*, **29** (2007), 79–91. <https://doi.org/10.1007/s10851-007-0027-4>



AIMS Press

© 2025 the Author(s), licensee AIMS Press. This is an open access article distributed under the terms of the Creative Commons Attribution License (<https://creativecommons.org/licenses/by/4.0>)