Mathematics Interdisciplinary Research **11** (3) (2026) 253 – 271

Original Scientific Paper

An Efficient Algorithm Based on Descent Direction for Nonsmooth Optimization and its Application in Image Restoration

Kimia Hoseinpour and Zohreh Akbari *

Abstract

In this paper, we propose a novel descent-based hybrid algorithm for solving nonsmooth box-constrained optimization problems. The method combines an ε-subdifferential approximation with heuristic local and global line search strategies. Theoretical analysis establishes the global convergence of the algorithm under Fritz John optimality conditions. To demonstrate the effectiveness of the proposed approach, it is applied to several standard grayscale test images for denoising purposes, addressing various types of noise, including Gaussian and salt-and-pepper noise. Moreover, extensive experiments and comparative analyses with existing denoising algorithms confirm the superior performance of our method in terms of restoration quality, particularly with respect to PSNR and SSIM metrics.

Keywords: Box-constrained optimization, Subdifferential approximation method, Line search technique, Image restoration.

2020 Mathematics Subject Classification: 90C59; 49M37.

How to cite this article
K. Hoseinpour and Z. Akbari, An efficient algorithm based on descent direction for nonsmooth optimization and its application in image restoration, *Math. Interdisc. Res.* **11** (3) (2026) 253-271.

* Corresponding author (E-mail: z.akbari@umz.ac.ir)
Academic Editor: Abbas Saadatmandi
Received 29 September 2025, Accepted 28 February 2026
DOI:10.22052/MIR.2026.257676.1545

1. Introduction

We consider the following box-constrained nonsmooth optimization problem:

$$\begin{aligned} &\min f(x) \\ \text{s.t.} \quad &a \leq x \leq b, \end{aligned} \tag{1}$$

where $f : \mathbb{R}^n \to \mathbb{R}$ is a nonsmooth and locally Lipschitz continuous function, and the vectors a and b denote the lower and upper bounds on the variables, respectively. Problems of this form frequently arise in various fields, including signal and image processing, machine learning, engineering design, and financial modeling [1–3]. A wide range of algorithms have been developed to solve the problem (1): Shen et al. developed Newton-type methods that achieve quadratic convergence and demonstrate high efficiency in solving problem (1) with the regularized objective function [4]. However, it is computationally expensive due to the need for second-order derivatives. Jia et al. reformulated the Lagrangian approach for constrained problems by employing multipliers, with the Karush-Kuhn-Tucker (KKT) conditions determining optimality [5]. Although, solving the resulting systems can be challenging, particularly in large-scale scenarios. Among first-order methods, projected gradient algorithms are widely used. These methods perform gradient descent steps and project the iterates back into the feasible set defined by the box constraints, which is computationally efficient when the projection is straightforward [6]. Unlike these methods, the proposed algorithm does not require differentiability ans rely on Hessian or inverse-Hessian approximations, and remains effective for nonsmooth and large-scale problems. The limited-memory BFGS method with box-constraints is a quasi-Newton approach that approximates the inverse Hessian using limited memory, making it well suited for large-scale problems [7]. Active-set methods identify the active constraints at each iteration and solve a series of simpler subproblems accordingly [8]. Interior-point methods traverse the interior of the feasible region using barrier functions to maintain feasibility during optimization [9]. Augmented Lagrangian methods combine the objective function with penalty terms and multipliers to better handle constraints, especially in complex scenarios [10]. A more recent approach, called box-constraint transformation, reformulates more general constraints into box constraints using variable transformations. This allows the use of efficient solvers in contexts such as Signorini contact problems [11]. Heuristic methods have gained significant attention for solving box-constrained optimization problems, especially when the objective is nonsmooth or derivative information is unavailable. One such method is the quasisecant approach, which estimates descent directions using the quasisecant method [12]. Bottou et al. have discussed genetic algorithms and pattern search techniques for constrained optimization, which are particularly useful in noisy or non-differentiable settings [13]. Mesh Adaptive Direct Search, proposed by Audet and Dennis, is another powerful derivative-free optimization method that guarantees convergence under mild assumptions, even for nonsmooth and discontinuous functions [14].

A particularly relevant application of box-constrained optimization is image restoration, which aims to recover high-quality images from degraded observations affected by noise, blur, or compression artifacts. The general image restoration model is given by

$$y = Ax + c,$$

with $x \in \mathbb{R}^{m\times n}$ denoting the unknown clean image, $y \in \mathbb{R}^{m\times n}$ being the noisy observation, A as a known blurring operator, and c as the additive noise. The goal is to find the original image x. A common formulation for finding the original image x is the following minimization problem:

$$\min_{0\leq x\leq 1} \frac{1}{2}\|Ax - y\|_2^2 + \lambda\varphi(x), \tag{2}$$

where the first term measures fidelity to the observed data, the parameter λ is a regularization parameter, and φ acts as a regularization term to stabilize the solution. Popular methods for solving such problems include projected gradient descent, proximal gradient algorithms, ADMM methods, interior-point methods, and the split Bregman technique [15–18].

In this work, we propose a method for solving problem (1). We apply the efficient proposed method to solve the nonsmooth box-constrained problem (2). The proposed algorithm is a descent-based hybrid approach, consisting of two main components: (i) the computation of a descent direction via subdifferential set approximation [19], and (ii) a heuristic line search procedure for step size selection that ensures feasibility within the box constraints [20]. The algorithm is capable of performing a global search and approximating the global minimum, while also providing a practical solution for large-scale nonsmooth problems.

Unlike projected gradient or quasi-Newton methods, our approach does not require smoothness or Hessian information, making it suitable for nonsmooth and large-scale problems. Compared to heuristic method, it maintains theoretical convergence guarantees while being computationally tractable. The integration of local and global search mechanisms within a trust-region-inspired framework distinguishes our method from existing subgradient or bundle-type approaches.

The remainder of this paper is structured as follows: Section 2 introduces preliminaries and basic notations. Section 3 introduces a new descent direction computation based on finite approximations of the ε-subdifferential, ensuring descent without requiring smoothness or second-order information. Section 4 establishes finite termination and global convergence of the proposed algorithm. It is rigorously proven that any accumulation point generated by the algorithm satisfies the Fritz John necessary optimality conditions under mild assumptions (local Lipschitz continuity and bounded level sets). Section 5 reports numerical results, with a particular focus on image restoration. Finally, Section 6 concludes the paper.

2. Preliminaries and basic notation

This section introduces key definitions and concepts from nonsmooth analysis, as outlined in [21].

Definition 2.1. A function $f : \mathbb{R}^n \to \mathbb{R}$ is said to be *locally Lipschitz continuous* at a point $x \in \mathbb{R}^n$ if there exist constants $K \geq 0$ and $\varepsilon > 0$ such that

$$|f(y) - f(z)| \leq K\|y - z\| \quad \forall y, z \in B(x; \varepsilon), \tag{3}$$

where $B(x; \varepsilon)$ denotes the open ball of radius ε centered at x.

Definition 2.2. The *subdifferential* of a convex function $f : \mathbb{R}^n \to \mathbb{R}$ at the point $x \in \mathbb{R}^n$ is defined as:

$$\partial_c f(x) = \left\{ \xi \in \mathbb{R}^n \mid f(x') \geq f(x) + \xi^\top (x' - x),\ \forall x' \in \mathbb{R}^n \right\}.$$

Each element $\xi \in \partial_c f(x)$ is called a *subgradient* of f at x.

Definition 2.3. Let $f : \mathbb{R}^n \to \mathbb{R}$ be locally Lipschitz continuous at $x \in \mathbb{R}^n$, and let $\varepsilon \geq 0$. The (Goldstein) *ε-subdifferential* of f at x is defined as:

$$\partial_\varepsilon f(x) = \mathrm{cl}\,\mathrm{conv}\left\{\partial f(y) \mid y \in B(x; \varepsilon)\right\}, \tag{4}$$

where cl denotes closure and conv denotes convex hull. Each $\xi \in \partial_\varepsilon f(x)$ is called an ε-subgradient of f at x.

Throughout this work, we use the Euclidean norm $\|\cdot\|$ unless stated otherwise. We now recall the necessary optimality conditions for nonsmooth problems, known as the Fritz John (FJ) conditions. These conditions will be used later to characterize the stationary points of the constrained optimization problem.

Theorem 2.4 (Fritz John Necessary Conditions [21])**.** *Let x^* be a local minimizer of the problem*

$$\min\{f(x) \mid g_i(x) \leq 0,\ i = 1, \ldots, m\},$$

where $f, g_i : \mathbb{R}^n \to \mathbb{R}$ are locally Lipschitz continuous functions at $x^* \in S = \{x \mid g_i(x) \leq 0,\ i = 1, \ldots, m\}$. Then there exist multipliers $\lambda_i \geq 0$, $i = 0, \ldots, m$, such that

$$\begin{aligned}
&\sum_{i=0}^{m} \lambda_i = 1, \\
&\lambda_i g_i(x^*) = 0 \quad \textit{for} \quad i = 1, \ldots, m, \\
&\mathbf{0} \in \lambda_0 \partial f(x^*) + \sum_{i=1}^{m} \lambda_i \partial g_i(x^*).
\end{aligned}$$

3. A new global method

In this section, we propose an efficient descent-based algorithm for solving nonsmooth box-constrained optimization problems. This method combines a subgradient-based descent direction strategy, as developed in [19], with a heuristic line search technique for step length selection, as outlined in [20]. The descent direction is determined by iteratively approximating the ε-subdifferential. The heuristic line search procedure is employed to determine a step length that remains within the box constraints. It is implemented in both local and global search variants.

3.1 Computing a descent direction

Let $\varepsilon \in [0,1]$ and $x \in \mathbb{R}^n$. Consider the following optimization problem:

$$\tilde{v} := \arg\min \left\{ \|v\|^2 \mid v \in \partial_\varepsilon f(x) \right\}. \tag{5}$$

Let $\tilde{d} := -\frac{\tilde{v}}{\|\tilde{v}\|}$. By Lebourg's Mean Value Theorem, we have:

$$f(x+\varepsilon\tilde{d}) - f(x) \leq \varepsilon\langle \xi, \tilde{d}\rangle \leq \varepsilon\langle \tilde{v}, \tilde{d}\rangle = -\varepsilon\|\tilde{v}\|,$$

where $\xi \in \partial_\varepsilon f(x)$. Therefore, $\tilde{v}$ is a valid descent direction. However, solving (5) exactly is often computationally intractable. To address this, $\partial_\varepsilon f(x)$ is approximated using a finite subset $W_k = \{v_1, v_2, \dots, v_k\} \subset \partial_\varepsilon f(x)$, and the minimization is performed over its convex hull:

$$\bar{v} := \arg\min \left\{ \|v\|^2 \mid v \in convW_k \right\}, \quad d := -\frac{\bar{v}}{\|\bar{v}\|}. \tag{6}$$

The quality of this approximation is evaluated using the following inequality:

$$f(x+\varepsilon d) - f(x) \leq -c\varepsilon\|\bar{v}\|, \quad c \in (0,1).$$

If this condition is not satisfied, the set W_k is updated by adding a new vector $v \in \partial_\varepsilon f(x)$ such that $v \notin convW_k$. Under mild assumptions, this iterative process converges in a finite number of steps [19]. The algorithm is stated as follows:

Algorithm 1 Computation of Descent Direction

1: The current point x, choose random $d_1 \in S_1$, and parameters: $\delta, c, \alpha \in [0,1]$.
2: **if** f is differentiable at $x + \alpha d_1$ **then** set $v := \nabla f(x + \alpha d_1)$.
3: **else** choose $v \in \partial f(x + \alpha d_1)$.
4: Set $W_1 := \{v\}$, $m := 1$.
5: Solve (6) to obtain $\bar{v}_m$.
6: **if** $\|\bar{v}_m\|^2 \leq \delta$ **then** stop.
7: **else** compute $d_{m+1} := -\bar{v}_m/\|\bar{v}_m\|$.
8: **if** $f(x + \alpha d_{m+1}) - f(x) \leq -c\alpha\|\bar{v}_m\|$ **then** stop.
9: Add $v_{m+1} \in \partial f(x)$ such that $v_{m+1} \notin convW_m$, set $W_{m+1} := W_m \cup \{v_{m+1}\}$, set $m = m + 1$, return to line 5.

In Algorithm 1, the initial descent direction d_1 is chosen randomly from the unit sphere $S_1 = \{y \in \mathbb{R}^n \mid \|y\| = 1\}$. This random selection serves several purposes: it avoids bias toward any particular direction, ensures a uniform coverage of possible directions, and provides a neutral starting point, especially important for nonsmooth or complex functions. Moreover, since the algorithm iteratively refines the descent direction, the precise choice of the initial direction is not critical, making a random unit vector a simple and computationally efficient choice.

3.2 A heuristic technique for line search

To determine the optimal step length, we recall two heuristic methods: local and global search techniques [20]. These methods are designed to iteratively adjust the step size during the optimization process, ensuring that each iteration makes sufficient progress toward finding the optimal solution. The local search technique

Algorithm 2 Heuristic Local Search Technique

1: Choose random $x^0 \in [a,b]$, $\varepsilon, \alpha_{\min} > 0$, $c_1 \in (0,1), c_2 \in (0, c_1]$, a ratio $\gamma \in (0,1)$, bundle size m_b, counter $k := 0$. Compute $\alpha_{\max} = \max_i \left\{\max(|x_i^k - a_i|, |x_i^k - b_i|)\right\}$.
2: Compute $\alpha_k := \gamma^k \alpha_{\max}$. **if** $\alpha_k < \alpha_{\min}$, **then** stop and return x^k.
3: Apply Algorithm 1 and obtain $\bar{v}_k$, direction d_k and iteration m.
4: **if** $\|\bar{v}_k\| \leq \varepsilon$ **then** stop.
5: **if** $m \geq m_b$
6: **else** update $x^{k+1} := x^k$ and set $k = k + 1$.
7: Perform backtracking line search:

$$\bar{h} = \arg \max_{0<h<1} \left\{ f(x^k + h\alpha_k d_k) - f(x^k) \leq -c_2 h \|\bar{v}\| \right\}.$$

8: Set $x^k := x^k + \bar{h} d_k$, return to line 2.

involves refining the step length by iteratively applying a backtracking procedure. Starting from an initial guess, the algorithm adjusts the step size based on the current solution and uses a set of conditions to ensure that each step moves in the right direction. This method is computationally efficient and helps find a reasonable step length in the neighborhood of the current solution. Now we are ready to state the global line search algorithm. The global search method, on the other hand, explores a wider range of possible step lengths by adjusting the search space and considering a broader set of solutions. This method is particularly useful when the local search has converged to a suboptimal solution, as it allows for a broader exploration of the solution space. The global search technique helps ensure that the optimization process does not get stuck in local minima, improving the chances of finding a global optimum.

Algorithm 3 Heuristic Global Search Technique

1: The sets S and T are empty, and the parameters are: bundle size m_b, integer $K > 0$, and counters $t := 0, k := 0$.
2: Apply Algorithm 2 and obtain local solution within $x^k \in [a, b]$.
3: Compute $\alpha_{\max} = \max_{i=1,\cdots,n} \{\max(|x_i^k - a_i|, |x_i^k - b_i|)\}$, choose $\alpha_{\min} \in (0, \alpha_{\max})$, set $\beta = (\alpha_{\min} - \alpha_{\max})/K$.
4: Compute $\alpha_k = \alpha_{\min} + k\beta$. **if** $\alpha_k > \alpha_{\max}$ **then** stop and output S, T.
5: Randomly choose $d_0 \in S_1$ and $t = 0$.
6: Compute $y^t = x^k + \alpha_k d^t$ and apply Algorithm 1 and obtain d_t, $\bar{v}$.
7: **if** $t < m_b$ and $||\bar{v}|| \leq \varepsilon$ **then** store current solution $y^t = x^k$, $f(y^t)$, update sets $S = S \cup \{y^t\}$, $T = T \cup \{y^t\}$, and $k = k + 1$, return to line 5.
8: **else** Set $d^{t+1} = -\frac{\bar{v}}{||\bar{v}||}$ and $t = t + 1$, return to line 4.

3.3 Minimization algorithm

In this subsection, we propose a new minimization algorithm for solving the problem (1) by combining the descent direction algorithm with heuristic local and global algorithms. In order to extend the quasisecant framework to the setting of global optimization, it is essential to introduce the concept of a trust region associated with a local minimizer. Before describing the algorithm, fundamental definitions are established.

Definition 3.1. Let x denote an isolated minimizer of a function f. A trust region of f at x is defined as a connected region $TR(x) \subset \mathbb{R}^n$ and has the property that every steepest descent trajectory of f, initiated from any point in $TR(x)$, converges to x. Conversely, no steepest descent trajectory starting from a point outside $TR(x)$ converges to x.

Suppose now that x is an isolated local minimizer of f. Consider the univariate restriction of f along an arbitrary direction $d \in S^1$: $z(\lambda) = f(x + \lambda d)$. Since x is locally minimizing, there exists a scalar $\lambda_0 > 0$ such that the function $z(\lambda)$ is strictly increasing on the interval $[0, \lambda_0]$. For a given direction d, we define the maximal admissible step size as $\lambda_{\max}(d) = \max\{\lambda_0 > 0 \mid z(\lambda) > 0,\ \forall \lambda \in [0, \lambda_0]\}$.

Definition 3.2. If x is an isolated local minimizer, the set

$$TR(x) = \bigcup_{d \in S_1} D(d),$$

is referred to as the trust region of x, where each directional segment is given by $D(d) = \{y \in \mathbb{R}^n \mid y = x + \lambda d,\ \lambda \in [0, \lambda_{\max}(d)]\}$. The point x is termed the *base point* of the trust region $TR(x)$.

Within the framework of global optimization, the principal objective is to identify the Trust Region whose base point corresponds to the global minimizer. Once trust regions are identified, they can be ranked by comparing the objective values of their base points. To further formalize this comparison, we introduce the following definition.

Definition 3.3. Let $TR(x_1)$ and $TR(x_2)$ denote two distinct trust regions of the function f, with base points x_1 and x_2, respectively. Then,

- $TR(x_1)$ is said to be lower than $TR(x_2)$ (equivalently, $TR(x_2)$ is higher than $TR(x_1)$) if $f(x_1) < f(x_2)$;
- $TR(x_1)$ and $TR(x_2)$ are said to be equivalent if $f(x_1) = f(x_2)$.

Algorithm 4 Global Subdifferential Minimization Method

1: Initialize with random $\tilde{x} \in \mathbb{R}^n$, set counter $k := 0$.
2: Apply Algorithm 2 by starting point $\tilde{x}$ to obtain x^k, set $x_{\text{opt}} = x^k$, $f_{\text{opt}} = f(x_{\text{opt}})$.
3: Apply Algorithm 3 by starting point x^k to generate sets $S = \{y_1, ..., y_K\}$, $T = \{f(y_1), ..., f(y_K)\}$, where K is the split parameter of Algorithm 3.
4: Let $\bar{y} = \arg\min_{y \in S} f(y)$.
5: **if** $f(\bar{y}) < f_{\text{opt}}$ Set $\tilde{x} := \bar{y}$, increment k, return to line 2.
6: Initialize $y_0 := x^k$ and evaluate $f(y_0)$. Starting with $m = 1$, examine each element in the set T sequentially, and form the index set

$$I = \{i \mid f(y_m) \leq f(y_{i-1}) \text{ and } f(y_i) \leq f(y_{i+1}),\ i = 1, 2, \ldots, K-1\}.$$

Next, refine the sets S and T by keeping only the elements whose indices belong to I:

$$S = \{y_i \in X \mid i \in I\}, \quad T = \{F(y_i) \in F \mid i \in I\}.$$

7: Use each $y \in S$ as a new starting point and apply Algorithm 2 to obtain a collection of local minimizers. Let $\bar{x}$ denote the point with the minimum objective function value among them. If $f(\bar{x}) < f_{opt}$, then update $x_{k+1} = \bar{x}$, $x_{opt} = \bar{x}$, $f_{opt} = f(x_{opt})$, $k := k+1$, return to line 3. Otherwise, terminate the algorithm and report x_{opt} and f_{opt}.

Algorithm 4 combines the local search (Algorithm 2) and the global search (Algorithm 3) to locate the global minimizer of a function, even when the function has multiple local minimum. Specially, Algorithm 2 refines candidate solutions, while Algorithm 3 identifies promising starting points for local searches in a trust region. In line 2, the local search explores the neighborhood using quasisecants, iteratively computing descent directions from approximated subgradients to find a local minimum. Its primary goal is to improve the current solution within a small region. In line 3, the global search is initiated from the obtained local minimizer, producing a set S of candidate starting points and a corresponding set T of function values. The split parameter K controls the density of the search in the surrounding area, allowing the algorithm to gradually expand the search region. The global search explores spheres of increasing radius around the local minimizer to identify new promising points. Unlike the local search, no line search is performed here, as the goal is exploration rather than refinement. In line 6, if a candidate point improves upon the current best solution, the algorithm restarts the local search from that point, enabling it to escape local minima and move toward potentially better solutions. In line 7, new promising trust regions are identified, and an index set is constructed to retain only candidates that may lead to further exploration. Line 5 provides a second opportunity to explore newly discovered trust regions. The algorithm iteratively restarts local searches from promising areas to find improved minima. If no further improvement is possible, the search terminates. The procedure ensures finite termination by counting the number of global iterations and dynamically updating whenever a better local minimum is discovered.

4. Convergence of the proposed algorithm

In this section, we establish the global convergence of Algorithm 4 for solving the problem (1). Specifically, we show that the generated solution x^* satisfies the Fritz John (FJ) necessary optimality conditions. To this end, we demonstrate that Algorithms 2 and 3 terminate after a finite number of iterations and that the resulting point x^* meets the optimality requirements. Global convergence is guaranteed under the following assumptions:

(A1) The objective function f is locally Lipschitz continuous.

(A2) The level set $M := \{x \in \mathbb{R}^n \mid f(x) \leq f(x_1)\}$ is bounded for a given point $x_1 \in \mathbb{R}^n$.

In [19], it is shown that if the sufficient decrease condition is not satisfied for the set $convW_k$, considered as an approximation of $\partial_\varepsilon f(x)$, then after a finite number of iterations, one can find a new ε-subgradient at the point x that does not belong to $convW_k$, which means that there exists v such that $v \notin convW_k$ thus $\|\bar{v}_k\|^2 \leq \langle v, \bar{v}_k\rangle = v^T\bar{v}_k$.

Proposition 4.1. *Consider a finite set* $W_k = \{v_1, v_2, ..., v_k\} \subset \partial_\varepsilon f(x)$, $0 \notin convW_k$ *and let*

$$\bar{v}_k := \arg\min\{\|v\|^2 \mid v \in convW_k\}.$$

Suppose that for the direction $d := -\frac{\bar{v}_k}{||\bar{v}_k||}$, *satisfy*

$$f(x + \varepsilon d) - f(x) > -c\varepsilon||\bar{v}_k||.$$

Then there exist $v \in \partial f(x + td)$ *and* $t \in [0, \varepsilon]$ *such that* $\langle v, \bar{v}_k\rangle \leq c||\bar{v}_k||^2$ *and* $v \notin convW_k$.

Proof. The proof is given in [19]. □

In Line 2 of Algorithms 2 and 4, which is applied, terminates in a finite number of iterations. Furthermore, it is proved in Theorem 4.2 that Algorithm 1, which is called within Algorithm 2, also terminates after finitely many iterations.

Theorem 4.2. *Algorithm 4 terminates in a finite number of iterations for any starting point.*

Proof. By Proposition 4.1, if the stopping condition is satisfied, then the subgradient $v \in convW_k$ is obtained from Equation (6). The stopping condition in line 3 of Algorithm 4 is defined as follows:

$$\|\bar{v}\|^2 \leq v_{k+1}^T \bar{v}, \quad \forall v \in convW_k.$$

We aim to find $v_{k+1} \in \partial f(x)$ such that $v_{k+1} \notin convW_k$, which ensures that

$$\langle v_{k+1}, \bar{v} \rangle = \|\bar{v}\|^2 \geq v_{k+1}^T \bar{v}, \quad \forall v \in convW_k.$$

Assume for some $c_1 \in (0, 1)$ that

$$v_{k+1}^T \bar{v} < c_1 \|\bar{v}\|^2. \tag{7}$$

Let L be the Lipschitz constant of f over the level set M. If (5) is not satisfied after finitely many iterations, then for some m, the norm $\|W_k\| \leq \delta$, and the algorithm terminates. For all $\theta \in [0, 1]$, we have:

$$\begin{aligned}\|\bar{v}_{k+1}\|^2 &\leq \|\theta v_{k+1} + (1-\theta)\bar{v}_k\|^2 \\ &\leq \|\bar{v}_k\|^2 + 2\theta \langle \bar{v}_k, v_{k+1} - \bar{v}_k \rangle + \theta^2 \|v_{k+1} - \bar{v}_k\|^2.\end{aligned}$$

Using inequality (7) and the fact that $\|v_{k+1} - \bar{v}_k\| \leq 2L$, we obtain:

$$\|\bar{v}_{k+1}\|^2 \leq \|\bar{v}_k\|^2 - 2\theta(1 - c_1)\|\bar{v}_k\|^2 + 4\theta^2 L^2. \tag{8}$$

Assume $\theta := (1 - c_1)(2L)^{-2}\|\bar{v}_k\|^2$, the parameter θ is chosen to minimize the quadratic upper bound (8), which is convex in θ.

Differentiating this expression and setting the derivative to zero yields the optimal value above, ensuring the tightest contraction estimate. Moreover, since $1 - c_1 > 0$ and $L > 0$, it follows that $\theta \in [0, 1]$, making it admissible in the preceding inequality. So:

$$\|\bar{v}_{k+1}\|^2 \leq \{1 - [(1 - c_1)(2L)^{-1}\|\bar{v}_k\|^2\}\|\bar{v}_k\|]^2.$$

Since $\|\bar{v}_k\| > \delta$, for any $k = 1, \cdots, m - 1$ and it obtains:

$$\|\bar{v}_{k+1}\|^2 \leq \{1 - [(1 - c_1)(2L)^{-1}\delta\|^2\}\|\bar{v}_k\|]^2.$$

Define the contraction factor:

$$L_1 := 1 - [(1 - c_1)(2L)^{-1}\delta]^2, \quad L_1 \in (0,1).$$

Then applying the contraction recursively gives:

$$\|\bar{v}_m\|^2 \le L_1\|\bar{v}_{m-1}\|^2 \le \cdots \le L_1^m\|\bar{v}_1\|^2 \le L_1^{m-1}L^2.$$

Since $|\bar{v}_k| \le L$ by definition of the level set and the Lipschitz constant, we further obtain $\|\bar{v}_m\|^2 \le L_1^{m-1}L^2$. Because $L_1 \in (0,1)$, the L_1^{m-1} ensuring that the expression on the right tends to 0 as $m \to \infty$. Therefore, the stopping condition $||\bar{v}_m||^2 \le \delta^2$ is satisfied if $L_1^{m-1}L^2 \le \delta^2$.

Thus, after at most finitely many steps, the inequality $||\bar{v}_k|| \le \delta$ becomes true, and at that point the algorithm terminates. In other words, the contraction ensures that the norm cannot stay above δ indefinitely, because it is multiplied by a factor strictly smaller than one each time. Hence, the algorithm terminates after finitely many iterations. □

We now reformulate problem (1) in standard inequality form:

$$\min f(x) \\ \text{s.t.} \quad \begin{cases} a - x \le 0, \\ x - b \le 0. \end{cases} \tag{9}$$

According to Theorem 2.4, if x^* is a local optimum of problem (9), then the Fritz John optimality conditions are given by:

$$\mathbf{0} \in \lambda_0 \partial f(x^*) + \begin{pmatrix} \lambda_1 \\ \vdots \\ \lambda_n \end{pmatrix} - \begin{pmatrix} \mu_1 \\ \vdots \\ \mu_n \end{pmatrix}, \tag{10}$$

$$\lambda_0 + \sum_{i=1}^{n}(\lambda_i + \mu_i) = 1, \tag{11}$$

where λ_0 and $\lambda_i, \mu_i \ge 0$ for $i = 1, \ldots, n$.

Theorem 4.3. *Algorithm 4 generates a point x^* satisfying the Fritz John conditions.*

Proof. From Theorem 4.2, since $\mathbf{0} \in \partial f(x^*)$, the set $\lambda_0 \partial f(x^*)$ contains the zero vector for any scalar λ_0(indeed $\lambda_0 \cdot \mathbf{0} = \mathbf{0}$). Hence, condition (10) simplifies to:

$$\lambda_i = \mu_i, \ \forall i = 1, 2, ..., n. \tag{12}$$

Combining (11) and (12), we obtain:

$$\lambda_0 + 2\sum_{i=1}^{n}\lambda_i = 1. \tag{13}$$

The system (13) that defined by (10), is feasible for nonnegative multipliers $\lambda_i, \mu_i \ge 0$, thereby verifying optimality. □

5. Numerical results

In this section, we evaluate the performance of the proposed algorithm by applying it to the problem of image restoration [22]. Specifically, we solve problem (2) using the anisotropic and directional total variation regularization functions as $\varphi(x)_{\text{atv}}$ and $\varphi(x)_{\text{dtv}}$ respectively, as follows:

$$\varphi(x)_{\text{atv}} = \sum_{i,j}(|x_{i+1,j} - x_{i,j}| + |x_{i,j+1} - x_{i,j}|),$$
$$\varphi(x)_{\text{dtv}} = \sum_{i,j}\sum_{(p,q)\in\mathcal{N}} w_{p,q}|x_{i+p,j+q} - x_{i,j}|,$$

where, $\mathcal{N} = \{(\pm1,0),(0,\pm1),(\pm1,\pm1)\}$ is eight-neighbor discrete set of directions.

To evaluate the effectiveness of our approach, we conducted experiments on several widely used grayscale test images, including *Cameraman*, *House*, *Lake*, *Lena*, and *Mandrill*, as referenced in [23]. Each image has a resolution of 256×256 pixels. For noise simulation, two types of degradation were applied: additive Gaussian noise with standard deviation $\sigma = \{0.3, 0.5, 0.7\}$, and salt-and-pepper noise with intensity $\gamma = \{10, 20\}$. The degradation matrix $A \in \mathbb{R}^{256\times256}$ is considered as the identity matrix.

Next, we implemented and compared the following algorithms for image denoising by using the $\varphi(x)_{\text{atv}}$ regularization: Algorithm 4 (Proposed Method), Adam Algorithm [24], Chambolle Algorithm [25], Limited-Memory-Gradient-Projection (LMGP) Algorithm [6].

The evaluation metrics include:

- **CPU Time** (in seconds): measures the computational cost.
- **PSNR** (Peak Signal-to-Noise Ratio):

$$\text{PSNR} = 10\cdot\log_{10}\left(\frac{MAX_I^2}{\text{MSE}}\right), \quad \text{MSE} = \frac{1}{mn}\sum_{i=1}^{m}\sum_{j=1}^{n}(x_{i,j} - y_{i,j})^2.$$

where $x_{i,j}$, and $y_{i,j}$ are pixel values at position (i,j) in the original and denoised image, respectively, $MAX_I = 255$ for 8-bit images, and mn is the total number pixels.

- **SSIM** (Structural Similarity Index):

$$\text{SSIM} = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2\mu_y^2 + C_1)(2\sigma_x^2 + \sigma_y^2 + C_2)},$$

PSNR evaluates restoration quality; higher values indicate better fidelity to the original image.

where μ_x, μ_y, σ_x^2, and σ_y^2 are mean intensity of original image x and denoised image y, variance of images x, y, respectively, the σ_{xy} is covariance between images x, y, and C_1, C_2 are small stabilizing constants to avoid division by zero.

The initialization parameters used across all methods were: $c_1 = 0.2$, $c_2 = 0.05$, $\lambda_{\min} = 0.01$, $\lambda = \alpha_{\min} = 0.1$, $m_b = n+3$, $K = n+1$, $\varepsilon = 10^{-7}$ and $iteration = 100$. Table 1 reports the SSIM values obtained by the proposed method and three bench-

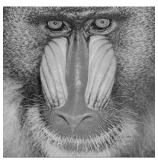
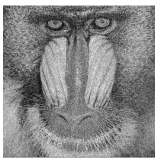
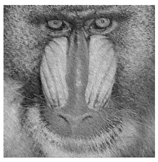
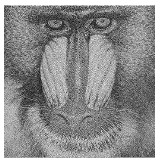
Table 1: SSIM comparison.

		Images				
		Cameraman	House	Lake	Lena	Mandrill
	Proposed Method	0.40	0.35	0.53	0.39	0.61
$\sigma = 0.5$	Adams Algorithm	0.36	0.33	0.50	0.35	0.57
$\gamma = 10$	Chambolle Algorithm	0.34	0.29	0.47	0.32	0.54
	LMGP Algorithm	0.38	0.34	0.51	0.34	0.56
	Proposed Method	0.44	0.38	0.57	0.43	0.64
$\sigma = 0.3$	Adams Algorithm	0.40	0.36	0.53	0.39	0.59
$\gamma = 20$	Chambolle Algorithm	0.37	0.33	0.50	0.36	0.57
	LMGP Algorithm	0.42	0.37	0.55	0.38	0.60
	Proposed Method	0.32	0.28	0.46	0.32	0.53
$\sigma = 0.7$	Adams Algorithm	0.32	0.29	0.46	0.31	0.51
$\gamma = 10$	Chambolle Algorithm	0.27	0.22	0.41	0.24	0.46
	LMGP Algorithm	0.29	0.27	0.44	0.27	0.48

mark algorithms on five standard test images under different noises (σ, γ). As can be observed, the proposed method consistently achieves the highest SSIM values for all test images. This indicates better structural preservation compared to Adams, Chambolle, and LMGP Algorithms. Lower noise level ($\sigma = 0.3$) leads to higher SSIM values for all methods. Higher noise level ($\sigma = 0.7$) degrades performance, but the Proposed Method remains more robust. Images with richer textures (*Mandrill*) consistently achieve higher SSIM values. Simpler images (*House*, *Cameraman*) show lower absolute SSIM, but the relative improvement of the Proposed Method remains clear. The Proposed Method provides an additional noticeable SSIM gain over both baselines in all cases. Table 2 summarizes the visual and quantitative results for each test image under different methods: Figure 1 further illustrates PSNR improvements for the images, highlighting the effectiveness of the proposed method. Subsequently, we implemented the Proposed Algorithm and compared the PSNR improvements in different $\varphi(x)_{\text{atv}}$ and $\varphi(x)_{\text{dtv}}$ regularizations. Figure 2 shows the results of the comparison for the *Cameraman*, *House*, *Lake*, *Lena* and *Mandrill* images.

The experimental results show that the proposed method consistently achieves

Table 2: PSNR and CPU time comparison with $\sigma = 0.5, \gamma = 10$.

Original Image	Noisy Image	Proposed Method	Adam Algorithm	Chambolle Algorithm
				
Cameraman		PSNR = 80.84 Time = 7.20	PSNR = 70.82 Time = 5.34	PSNR = 58.75 Time = 4.53
				
House		PSNR = 77.33 Time = 9.11	PSNR = 68.02 Time = 6.22	PSNR = 55.66 Time = 4.81
				
Lake		PSNR = 78.36 Time = 6.73	PSNR = 67.93 Time = 4.95	PSNR = 56.72 Time = 4.06
				
Lena		PSNR = 80.34 Time = 6.92	PSNR = 70.70 Time = 5.03	PSNR = 58.68 Time = 4.25
				
Mandrill		PSNR = 80.33 Time = 8.09	PSNR = 69.48 Time = 4.97	PSNR = 58.85 Time = 3.95

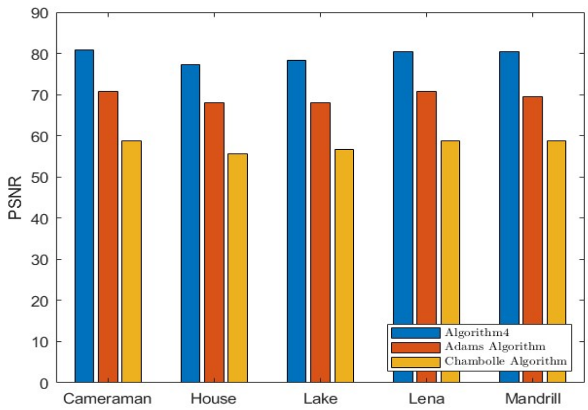


Figure 1: PSNR comparison results for different algorithm by $\sigma = 0.5, \gamma = 10$.

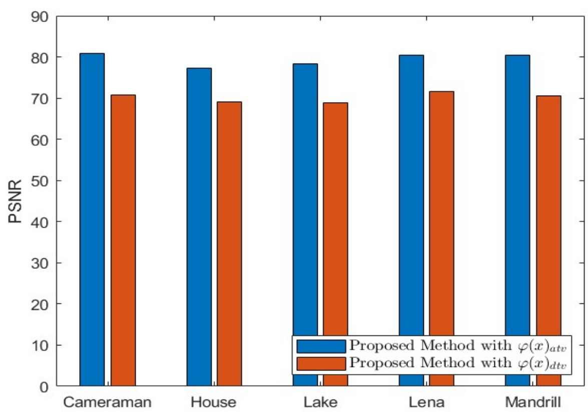


Figure 2: PSNR comparison results for different regularizations by $\sigma = 0.5, \gamma = 10$.

higher SSIM and PSNR values compared to Adam, Chambolle, and LMGP algorithms across all test images, confirming its superior denoising performance. While the computational cost is moderately higher, the improvement in restoration quality justifies the trade-off. Notably, in images with complex textures (e.g., *Mandrill*), our method is significantly better at preserving fine details where traditional methods fail. The $\varphi(x)_{\mathrm{dtv}}$ function demonstrates inferior performance compared to $\varphi(x)_{\mathrm{atv}}$ in terms of image restoration PSNR. This can be attributed to $\varphi(x)_{\mathrm{dtv}}$'s sensitivity to both the method of weight assignment and the specific weight values themselves. It is expensive computationally and has possible geometric mismatches. Minor variations in these parameters lead to significant changes in the restoration outcome. Consequently, $\varphi(x)_{\mathrm{atv}}$ is a more suitable choice, yielding more acceptable and reliable image restoration results.

6. Conclusions

In this work, we proposed a novel approach for solving nonsmooth box-constrained optimization problems, with a focus on image restoration. The method integrates a hybrid descent direction strategy with heuristic line search techniques, providing an effective solution for denoising images corrupted by various types of noise. Extensive experiments on standard grayscale images demonstrate the robustness of the proposed approach, outperforming existing algorithms in both restoration quality and computational efficiency.

Future work will aim to extend this framework to generalized nonsmooth optimization with box constraints, addressing more complex and large-scale problems such as multi-modal image restoration and other signal processing tasks, and integration with stochastic. Furthermore, we plan to combine the heuristic strategy with proximal gradient methods to further enhance performance. Another purpose is to combine the proposed descent direction with modern learning-based or hybrid variational deep approaches. For example, integrating algorithm-driven priors with neural network-based regularizer or using the proposed method as an optimization layer inside deep image restoration architectures could potentially improve both robustness and computational efficiency. Analyzing the theoretical properties of such hybrid models within the nonsmooth optimization setting would also be an interesting avenue for further research.

Conflicts of Interest. The authors declare that they have no conflicts of interest regarding the publication of this article.

References

[1] M. Gaudioso, G. Giallombardo and G. Miglionico, Essentials of numerical nonsmooth optimization, *Ann. Oper. Res.* **314** (2022) 213 – 253,

https://doi.org/10.1007/s10479-021-04498-y.

[2] B. Acharjee, M. H. Hanif and O. Waqar, Deep unsupervised learning for optimization with box and monotone matrix based polytope constraints: A case-study of D2D wireless networks, *IEEE Wirel. Commun. Lett.* **12** (2023) 2223 − 2227, https://doi.org/10.1109/LWC.2023.3316114.

[3] L. Bottou, F. E. Curtis and J. Nocedal, Optimization methods for large-scale machine learning, *SIAM Rev.* **60** (2018) 223 − 311, https://doi.org/10.1137/16M1080173.

[4] C. Shen, W. Xue, L-H. Zhang and B. Wang, An active-set proximal-Newton algorithm for ℓ_1 regularized optimization problems with box-constraints, *J. Sci. Comput.* **85** (2020) #57, https://doi.org/10.1007/s10915-020-01364-0.

[5] X. Jia, C. Kanzow, P. Mehlitz and G. Wachsmuth, An augmented Lagrangian method for optimization problems with structured geometric constraints, *Math. Program.* **199** (2023) 1365−1415, https://doi.org/10.1007/s10107-022-01870-z.

[6] S. Crisci, F. Porta, V. Ruggiero and L. Zanni, Hybrid limited memory gradient projection methods for box-constrained optimization problems, *Comput. Optim. Appl.* **84** (2023) 151−189, https://doi.org/10.1007/s10589-022-00409-4.

[7] A. Bemporad, An L-BFGS-B approach for linear and nonlinear system identification under ℓ_1 and group-lasso regularization, *IEEE Trans. Autom. Control* **70** (2025) 4857 − 4864, https://doi.org/10.1109/TAC.2025.3541018.

[8] A. Neumaier, B. Azmi and M. Kimiaei, An active set method for bound-constrained optimization, *Optim. Methods Softw.* **39** (2024) 1216 − 1240, https://doi.org/10.1080/10556788.2024.2339215.

[9] Y. Yang, An efficient arc-search interior-point algorithm for convex quadratic programming with box-constraints, *Numer. Algorithms* **91** (2022) 711 − 748, https://doi.org/10.1007/s11075-022-01279-x.

[10] Z. Y. Wu, F. S. Bai and J. Tian, Optimization methods for box-constrained nonlinear programming problems based on linear transformation and Lagrange interpolating polynomials, *J. Oper. Res. Soc. China* **5** (2017) 193−218, https://doi.org/10.1007/s40305-017-0157-3.

[11] T. Takahashi and C. Batty, Optimizing parameters for static equilibrium of discrete elastic rods with active-set Cholesky, *IEEE Trans. Vis. Comput. Graph.* **32** (2026) 1951 − 1962, https://doi.org/10.1109/TVCG.2025.3622483.

[12] W. Cheng, Z. LinPeng and D. Li, An inexact quasi-Newton algorithm for large-scale ℓ_1 optimization with box constraints, *Appl. Numer. Math.* **193** (2023) 179 − 195, https://doi.org/10.1016/j.apnum.2023.07.004.

[13] L. Bottou, F. E. Curtis and J. Nocedal, Optimization methods for large-scale machine learning, *SIAM Rev.* **60** (2018) 223 − 311.

[14] C. Audet and J. E. Jr. Dennis, Mesh adaptive direct search algorithms for constrained optimization, *SIAM J. Optim.* **17** (2006) 188 − 217, https://doi.org/10.1137/040603371.

[15] E. Chouzenoux, M. C. Corbineau and J. C. Pesquet, A proximal interior point algorithm with applications to image processing, *J. Math. Imaging Vision* **62** (2020) 919 − 940, https://doi.org/10.1007/s10851-019-00916-w.

[16] T. Tirer and R. Giryes, On the convergence rate of projected gradient descent for a back-projection based objective, *SIAM J. Imaging Sci.* **14** (2021) 1504−1531, https://doi.org/10.1137/21M1407902.

[17] W. Wang, C. Wu and X. C. Tai, A globally convergent algorithm for a constrained non-Lipschitz image restoration model, *J. Sci. Comput.* **83** (2020) #14, https://doi.org/10.1007/s10915-020-01190-4.

[18] W. S. Xie, Y. F. Yang and B. Zhou, An ADMM algorithm for second-order TV-based MR image reconstruction, *Numer. Algorithms*, **67** (2014) 827−843, https://doi.org/10.1007/s11075-014-9826-z.

[19] N. Mahdavi-Amiri and R. Yousefpour, An effective nonsmooth optimization algorithm for locally Lipschitz functions, *J. Optim. Theory Appl.* **155** (2012) 180 − 195, https://doi.org/10.1007/s10957-012-0024-7.

[20] Q. Long, C. Wu, X. Wang and Z. Wu, A modifed quasisecant method for global optimization, *Appl. Math. Model.* **51** (2017) 21 − 37.

[21] A. Bagirov, N. Karmitsa and M. M. Mäkelä, *Introduction to Nonsmooth Optimization, Theory, Practice and Software*, Springer Cham, 2014.

[22] K. Hoseinpour and Z. Akbari, (2025). MATLAB code [Computer software]. GitHub. Available at: https://github.com/kimiahoseinpour/matlab-code.git

[23] A. Weber, Signal and I. P. I. of the University of Southern California, The USCSIPI image database, http://sipi.usc.edu/database/ **8** (1997).

[24] D. P. Kingma and J. Ba, Adam: A method for stochastic optimization, *arXiv:1412.6980* (2014).

[25] A. Chambolle, An algorithm for total variation minimization and applications, *J. Math. Imaging Vision*, **20** (2004) 89 − 97, https://doi.org/10.1023/B:JMIV.0000011325.36760.1e.

Kimia Hoseinpour
Department of Applied Mathematics,
Faculty of Mathematical Sciences,
University of Mazandaran,
Babolsar, I. R. Iran
e-mail: k.hosseinpour01@umail.umz.ac.ir

Zohreh Akbari
Department of Applied Mathematics,
Faculty of Mathematical Sciences,
University of Mazandaran,
Babolsar, I. R. Iran
e-mail: z.akbari@umz.ac.ir