AIMS Mathematics, 10(8): 18603–18626.
DOI:10.3934/math.2025831
Received: 22 April 2025
Revised: 24 July 2025
Accepted: 04 August 2025
Published: 18 August 2025

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

Research article

Accelerated over-relaxation heavy-ball hard thresholding pursuit for compressive sensing

Xinyu Diao[1], **Zhongfeng Sun**[1], **Jingyong Tang**[2] **and Jinchuan Zhou**[1,*]

[1] School of Mathematics and Statistics, Shandong University of Technology, Zibo 255000, China
[2] School of Mathematics and Statistics, Xinyang Normal University, Xinyang 464000, China

* **Correspondence:** Email: jinchuanzhou@163.com.

Abstract: In this paper, we propose a novel algorithm, called Accelerated Over-Relaxation Heavy-Ball Hard Threshold Pursuit (AOR-HBHTP), for solving compressive sensing problems. The algorithm incorporates the Accelerated Over-Relaxation technique and Heavy-Ball momentum into the Hard Threshold Pursuit framework. Theoretical results include establishing convergence analysis and providing an estimation of the number of iteration steps. We show that, as long as the measurement matrix satisfies the restricted isometry property, AOR-HBHTP can successfully recover unknown signals within a number of iterations proportional to the sparsity level. The upper bound on the number of iterations is uniform in the sense that it does not depend on any unknown special-signal information. In numerical experiments, we evaluate recovery capability, success rate, and runtime of AOR-HBHTP by using Phase Transition Curve, Algorithm Selection Map, and Signal-to-Noise Ratio. The promising numerical results demonstrate the effectiveness of AOR-HBHTP in recovering sparse signals.

Keywords: compressive sensing; sparse signal recovery; restricted isometry property; hard thresholding pursuit; accelerated over-relaxation heavy-ball methods
Mathematics Subject Classification: 90C26, 90C52

1. Introduction

The compressive sensing problem aims to recover a sparse signal $x \in \mathbb{R}^N$ from the following linear measurement:

$$y = Ax + e, \tag{1.1}$$

where $A \in \mathbb{R}^{m\times N}$ represents the measurement matrix, and e denotes the measurement error. A fundamental requirement in compressive sensing is that m (the number of measurements) is much smaller than N (the length of signals). Thus, problem (1.1) becomes an underdetermined system of

linear equations. To recover the unknown signal from the above measurement, it is typically necessary to impose a special structure on the signal, such as sparsity. This leads to the following mathematical model for solving (1.1)

$$\min f(x) := \frac{1}{2}\|y - Ax\|^2, \quad \text{s.t. } \|x\|_0 \leq s, \tag{1.2}$$

where s is a predefined sparsity degree, and $\|x\|$ and $\|x\|_0$ denote the 2-norm and the zero-norm, respectively. Since the zero-norm represents the number of nonzero elements in the vector x, the constraint $\|x\|_0 \leq s$ indicates that x is an s-sparse vector. Due to the presence of the zero-norm, problem (1.2) is both nonsmooth and nonconvex, which creates significant challenges in designing numerical methods for solving (1.2).

The widely-used algorithms for solving compressive sensing problems can be roughly categorized into three groups, including optimization methods, greedy methods, and thresholding-based methods; see [1–3] and references therein for more information. We first recall a simple yet highly effective hard thresholding algorithm, known as Hard Threshold Pursuit (HTP [4]), which enhances the Iterative Hard Thresholding (IHT [5]) algorithm by incorporating an orthogonal projection strategy. The iterative steps of HTP are given as follows:

$$\begin{cases} u^{n+1} := x^n - \alpha \nabla f(x^n), & \text{(1.3a)} \\ S^{n+1} := \mathcal{L}_s(u^{n+1}), & \text{(1.3b)} \\ x^{n+1} := \arg\min\{\|y - Az\| : \text{supp}(z) \subseteq S^{n+1}\}, & \text{(1.3c)} \end{cases}$$

where $\mathcal{L}_s(u^{n+1})$ denotes the set of indices corresponding to the s largest entries of u^{n+1}, and $\text{supp}(x)$ represents the support set of x, defined as $\text{supp}(x) := \{i \,|\, x_i \neq 0, i = 1, \ldots, N\}$. In the above iterative scheme, the first step (1.3a) computes a descent direction, the second step (1.3b) is used to ensure feasibility by selecting the support set, and the third step (1.3c) performs an unbiased optimization to find a more suitable vector over the index set obtained in step (1.3b). In the above algorithm framework, the sparsity level s is assumed to be known a priori. However, several researchers have investigated scenarios where the sparsity level is unknown; see [6–8]. Other related work includes hyperinterpolation [9, 10], non-negative sparse signal recovery [11], tensor completion [12–14], low-rank matrix recovery [15], and non-convex approximation methods [16, 17], etc.

Recently, first-order methods, which rely solely on gradient information without involving any second-order derivatives, have garnered significant attention due to their high efficiency and adaptability in solving large-scale data-driven applications. Their appeal lies in low computational cost while achieving high performance. However, the classic gradient descent method exhibits sawtooth oscillations when solving ill-posed problems, leading to very slow convergence speeds. To overcome this drawback, various acceleration techniques have been introduced, such as Heavy-ball methods [18–20] and Nesterov accelerated gradient methods [21–23], among others. The well-known heavy-ball update rule is given by

$$u^{n+1} = x^n - \alpha \nabla f(x^n) + \gamma(x^n - x^{n-1}). \tag{1.4}$$

The key difference between (1.3a) and (1.4) is the addition of a momentum term $x^n - x^{n-1}$. This modification enables Heavy-ball (HB) methods to update the iterative process by storing and utilizing information from previous points.

Global convergence properties of HB methods have been well-established; see [19, 24, 25]. For a function that is μ-strong convex and L-smooth, the convergence rate of HB methods for solving $\min f(x)$ is of the order $1 - O(\mu/L)$ [26], which is the same as that of gradient descent. In other words, although HB methods have demonstrated performance benefits over standard gradient-based methods, their theoretical convergence rate remains unchanged. Goujaud et al. [27] showed that HB method fails to achieve an accelerated convergence rate even for smooth and strongly convex optimization problems. Therefore, to improve convergence speed, we have to make appropriate modifications to HB methods beyond simply adjusting the coefficients in (1.4). More recently, a significant breakthrough in this area was achieved in [28], where the authors propose a variant of HB in the form of

$$x^{n+1} = x^n - \alpha(2\nabla f(x^n) - \nabla f(x^{n-1})) + \gamma(x^n - x^{n-1}). \tag{1.5}$$

Here, an over-relaxation strategy is applied by introducing a perturbation $\nabla f(x^n) - \nabla f(x^{n-1})$ to $\nabla f(x^n)$; i.e., the gradient term is modified from $\nabla f(x^n)$ to its approximation $2\nabla f(x^n) - \nabla f(x^{n-1})$. For this reason, the authors refer to the above method (1.5) as the Accelerated Over-Relaxation Heavy-Ball (AOR-HB) method. They prove that this modification successfully enables the algorithm to achieve the optimal linear convergence rate of $1 - O(\sqrt{\mu/L})$. Furthermore, by discretizing an ordinary differential equation known as the HB-saddle flow, the AOR-HB algorithm is extended from solving convex programming problems to a special class of nonconvex programming problems, specifically, min-max problems with bilinear coupling, while maintaining optimal iteration complexity.

A question arises: As a significant theoretical development in heavy-ball type algorithms, can the AOR-HB method be further extended to solve other nonconvex optimization problems? Our primary target of this paper is to provide an affirmative answer by applying AOR-HB to solve compressive sensing problems (1.2), which are nonconvex and nonsmooth due to the presence of the zero-norm. Taking into account the special structure of compressive sensing problems (1.2), it needs to incorporate a Hard Thresholding Index Operator $\mathcal{L}_s$ to ensure the sparsity of iteration points. For simplicity, we just present the iterative active set in our algorithms as follows:

$$\mathcal{S}^{n+1} = \mathcal{L}_s(x^n - \alpha\nabla f(x^n) - \beta\underbrace{(\nabla f(x^n) - \nabla f(x^{n-1}))}_{over-relaxation} + \gamma\underbrace{(x^n - x^{n-1})}_{heavy-ball}). \tag{1.6}$$

In (1.6), two types of momentum terms are combined to correct the gradient descent direction: A gradient-based term (the first term $\nabla f(x^n) - \nabla f(x^{n-1})$, referred to as over-relaxation momentum), and a variable-based momentum term (the second term $x^n - x^{n-1}$, referred to as heavy-ball momentum). If we set $\alpha = \beta$, the search direction in (1.6) coincides with (1.5); if $\beta = 0$, our algorithm reduces to the Heavy-ball Hard Thresholding Pursuit (HBHTP [29]). The parameters used in various algorithms mentioned above are summarized in Table 1.

Table 1. Comparison of parameters in different first-order methods.

Algorithms	Step-size: α	Over-Relaxation: β	Heavy-Ball: γ	Hard Thresholding: $\mathcal{L}_s$
Gradient Descent	✓	✗	✗	✗
HB	✓	✗	✓	✗
AOR-HB	✓	✓	✓	✗
HBHTP	✓	✗	✓	✓
AOR-HBHTP	✓	✓	✓	✓

The contributions of this paper are twofold: (i) Design a novel algorithm called AOR-HBHTP by incorporating the Accelerated Over-Relaxation technique and Heavy-Ball momentum into the Hard Threshold Pursuit framework; and (ii) establish a signal-independent upper bound on the number of iterations. Main works include the following aspects.

i) Theory analysis. We establish the global convergence of AOR-HBHTP under the assumption that the measurement matrix satisfies the restricted isometry property (RIP). We further obtain an estimate of the number of iterative steps. The upper bound on the number of iterations is proportional to the sparsity level, and hence has a uniform property in the sense that it does not depend on any special signal information. This result is new even for HBHTP, because the estimation obtained in [29] requires prior knowledge of the size of the signal to be recovered, which restricts its practical applicability because the target signal's information is typically unknown in advance.

ii) Numerical experiments. We compare AOR-HBHTP with six other popular algorithms: IHT, HTP, Compressive Sampling Matching Pursuit (CoSaMP [30]), Subspace Pursuit (SP [31]), HBHTP [29] and Enhanced Dynamic Orthogonal Matching Pursuit (EDOMP [32]) in both synthetic and real-world signals. Our numerical results demonstrate that, regardless of the presence of noise, AOR-HBHTP exhibits significant advantages in both recovery accuracy and runtime. Specifically, the added over-relaxation term enhances the algorithm's robustness by mitigating the issue of excessive oscillation during the iterative process. The observations from Phase Transition Curve (PTC), Algorithm Section Map (ASM), and Signal-to-Noise Ratio (SNR) indicate that both the over-relaxation technique and the heavy-ball momentum are effective acceleration strategies to improve the numerical performance of algorithms.

The notations used in this paper are standard. For a given index set $S \subset \{1, 2, \ldots, n\}$, let $|S|$ denote the cardinality of S, and let $\bar{S}$ denote the complement of S. For a fixed vector $x \in \mathbb{R}^N$, x_S is obtained by retaining the elements of x indexed in S and setting the remaining elements to zero. The transpose of a matrix A is denoted as A^T. For a scalar t, $\lfloor t \rfloor$ denotes the largest integer that does not exceed t. For two sets C and D, denote by $C \triangle D := (C \backslash D) \cup (D \backslash C)$ their symmetric difference. The structure of this paper is as follows: The proposed AOR-HBHTP algorithm is introduced in Section 2. In Section 3 we discuss the convergence analysis and address the estimation of iteration steps in Section 4. Numerical experiments are reported in Section 5. Conclusions are drawn Section 6.

2. Algorithm: AOR-HBHTP

Algorithm 1 Heavy-ball Hard Thresholding Pursuit (HBHTP).

Input: A measurement matrix A, a measurement vector y, a sparsity level s, and two nonnegative parameters α, γ. The iterative steps are given as follows:

(1) Initialization: Start from two s-sparse vectors $x^0, x^1 \in \mathbb{R}^n$. Typically $x^0 = x^1 = 0$.
(2) Iteration: Repeat the following steps until a stopping criterion is met.

$$\begin{aligned} &\text{Gradient step:} && u^{n+1} := x^n + \alpha A^T(y - Ax^n) + \gamma(x^n - x^{n-1}), \\ &\text{Support selection:} && S^{n+1} := \mathcal{L}_s(u^{n+1}), \\ &\text{Projection step:} && x^{n+1} := \arg\min_z \{\|y - Az\| : \operatorname{supp}(z) \subseteq S^{n+1}\}. \end{aligned}$$

Output: The s-sparse vector $\bar{x}$.

As shown above, HBHTP involves two parameters α and γ. When γ is set to zero, HBHTP reduces to the classical HTP. To enhance the performance of HBHTP, we incorporate not only the gradient information of the current point x^n but also that of the previous point x^{n-1}.

Algorithm 2 Accelerated Over-Relaxation Heavy-Ball Hard Thresholding Pursuit (AOR-HBHTP).

Input: A measurement matrix A, a measurement vector y, a sparsity level s, and three nonnegative parameters α, β, γ. The iterative steps are given as follows:

(1) **Initialization**: Start from two s-sparse vectors $x^0, x^1 \in \mathbb{R}^n$, typically $x^0 = x^1 = 0$.
(2) **Iteration**: Repeat the following steps until a stopping criterion is met.

$$\begin{aligned}
&\text{Gradient step:} && u^{n+1} := x^n + \alpha A^T(y - Ax^n) - \beta\left[A^T(Ax^n - y) - A^T(Ax^{n-1} - y)\right] + \gamma(x^n - x^{n-1}),\\
&\text{Support selection:} && S^{n+1} := \mathcal{L}_s(u^{n+1}),\\
&\text{Projection step:} && x^{n+1} := \arg\min_z \{\|y - Az\| : \mathrm{supp}(z) \subseteq S^{n+1}\}.
\end{aligned}$$

Output: The s-sparse vector $\bar{x}$.

3. Convergence analysis

In this section, we analyze convergence property of the AOR-HBHTP algorithm using the Restricted Isometry Property (RIP), one of the important analytical tools for studying theories and algorithms in the field of compressive sensing. To begin, we first recall the definition of RIP.

Definition 3.1. *[4] The restricted isometry constant (RIC) of order s for a measurement matrix $A \in \mathbb{R}^{m\times N}$ with $m \ll N$, denoted by δ_s, is defined as the smallest number $\delta \geq 0$ such that*

$$(1 - \delta)\|z\|^2 \leq \|Az\|^2 \leq (1 + \delta)\|z\|^2,$$

for every s-sparse vector $z \in \mathbb{R}^N$ (i.e., $\|z\|_0 \leq s$). If such a constant δ_s exists, matrix A is said to satisfy the restricted isometry property (RIP) of order s.

It is easy to see that $\delta_t \leq \delta_s$, whenever $t \leq s$; see [2]. Several basic inequalities related to RIP will be frequently utilized in the subsequent convergence analysis of AOR-HBHTP.

Lemma 3.1. *[4] Let $v \in \mathbb{R}^N$, $u \in \mathbb{R}^m$, $U \subseteq \{1, 2, ..., N\}$ be an index set, and t be a positive integer. Then, the following inequalities hold:*

(1) $\|((I - A^TA)v)_U\| \leq \delta_t\|v\|$, whenever $|\, U \cup supp(v) \,| \leq t$.
(2) $\|(A^Tu)_U\| \leq \sqrt{1 + \delta_t}\|u\|$, whenever $|\, U \,| \leq t$.

Lemma 3.2. *[29] Suppose that the non-negative sequence $\{w^n\}$ for $n = 0, 1, ...$ satisfies*

$$w^{n+1} \leq \widehat{b}w^n + \widehat{c}w^{n-1} + \widehat{d},$$

where $\widehat{b}, \widehat{c}, \widehat{d} \geq 0$ and $\widehat{b} + \widehat{c} < 1$. Then,

$$w^n \leq \theta^{n-1}[w^1 + (\theta - \widehat{b})w^0] + \frac{\widehat{d}}{1 - \theta},$$

with $\theta := \frac{\widehat{b} + \sqrt{\widehat{b}^2 + 4\widehat{c}}}{2} < 1$.

With these preparations, we are now in a position to prove the convergence of AOR-HBHTP.

Theorem 3.1. *Suppose that the RIC, δ_{3s}, of the measurement matrix A satisfies*

$$\delta_{3s} < \frac{1}{\sqrt{3}} \approx 0.577.$$

Choose parameters α, β, γ such that one of the following two cases

$$\begin{cases} (I): & \frac{\frac{1}{\widehat{\eta}}-\delta_{3s}}{1+\delta_{3s}} \le \beta < \frac{\frac{1}{\widehat{\eta}}-\delta_{3s}}{2\delta_{3s}}, & \frac{\delta_{3s}-\frac{1}{\widehat{\eta}}+(1+\delta_{3s})\beta}{1-\delta_{3s}} < \gamma < \beta, & \frac{1+2\beta\delta_{3s}-\frac{1}{\widehat{\eta}}}{1-\delta_{3s}} < \alpha < \frac{\frac{1}{\widehat{\eta}}+1-2(\delta_{3s}+1)\beta+2\gamma}{1+\delta_{3s}}, \\ (II): & 0 \le \beta < \frac{\frac{1}{\widehat{\eta}}-\delta_{3s}}{2\delta_{3s}}, & \beta < \gamma < \frac{-\delta_{3s}+\frac{1}{\widehat{\eta}}+(1-\delta_{3s})\beta}{1+\delta_{3s}}, & \frac{1+2\beta\delta_{3s}-\frac{1}{\widehat{\eta}}+2\gamma-2\beta}{1-\delta_{3s}} < \alpha < \frac{\frac{1}{\widehat{\eta}}+1-2\beta\delta_{3s}}{1+\delta_{3s}}, \end{cases} \tag{3.1}$$

where $\widehat{\eta} := \sqrt{\frac{2}{1-\delta_{3s}^2}}$. Then, the iterative sequence $\{x^n\}$ generated by AOR-HBHTP with $y = Ax + e$ satisfies

$$\|x^n - x_S\|_2 \le \theta^{n-1}\Big(\|x^1 - x_S\| + (\theta - \widehat{b})\|x^0 - x_S\|\Big) + \widehat{\tau}\|e'\|_2, \tag{3.2}$$

where $S := \mathcal{L}_s(x)$, $\theta := \frac{\widehat{b}+\sqrt{\widehat{b}^2+4\widehat{c}}}{2}$, $\widehat{b} := \widehat{\eta}(|1-\alpha-\beta+\gamma| + (\alpha+\beta)\delta_{3s})$, $\widehat{c} := \widehat{\eta}(\beta\delta_{3s} + |\beta-\gamma|)$, $\widehat{\tau} := \frac{1}{1-\theta}\left(\alpha\widehat{\eta}\sqrt{1+\delta_{2s}} + \frac{\sqrt{1+\delta_s}}{1-\delta_{2s}}\right)$, and $e' := Ax_{\bar{S}} + e$.

Proof. According to [4], we know

$$\|x^{n+1} - x_S\| \le \frac{1}{\sqrt{1-\delta_{2s}^2}}\|(x^{n+1} - x_S)_{\overline{S^{n+1}}}\| + \frac{\sqrt{1+\delta_s}}{1-\delta_{2s}}\|e'\|. \tag{3.3}$$

Since $S^{n+1} = \mathcal{L}_s(u^{n+1})$, then

$$\|(u^{n+1})_S\|^2 \le \|(u^{n+1})_{S^{n+1}}\|^2.$$

Simultaneously eliminating $\|(u^{n+1})_{S\cap S^{n+1}}\|^2$ from both sides of this inequality yields

$$\|(u^{n+1})_{S\setminus S^{n+1}}\| \le \|(u^{n+1})_{S^{n+1}\setminus S}\|. \tag{3.4}$$

For (3.4), the left-hand side is

$$\begin{aligned} \|(u^{n+1})_{S\setminus S^{n+1}}\| &= \|(u^{n+1} - x_S + x_S - x^{n+1})_{S\setminus S^{n+1}}\| \\ &\ge \|(x^{n+1} - x_S)_{S\setminus S^{n+1}}\| - \|(u^{n+1} - x_S)_{S\setminus S^{n+1}}\| \\ &= \|(x^{n+1} - x_S)_{\overline{S^{n+1}}}\| - \|(u^{n+1} - x_S)_{S\setminus S^{n+1}}\|, \end{aligned}$$

and the right-hand side is

$$\|(u^{n+1})_{S^{n+1}\setminus S}\| = \|(u^{n+1} - x_s)_{S^{n+1}\setminus S}\|.$$

Therefore, the inequality (3.4) can be rewritten as

$$\|(x^{n+1} - x_s)_{\overline{S^{n+1}}}\| - \|(u^{n+1} - x_s)_{S\setminus S^{n+1}}\| \le \|(u^{n+1} - x_s)_{S^{n+1}\setminus S}\|. \tag{3.5}$$

Note that $y = Ax + e = A(x_S + x_{\bar{S}}) + e = Ax_S + e'$. Hence,

$$u^{n+1} - x_S = x^n - x_S + \alpha A^T A(x_S - x^n) - \beta A^T A(x^n - x^{n-1}) + \gamma(x^n - x^{n-1}) + \alpha A^T(Ax_{\bar{S}} + e)$$

$$=(1-\alpha-\beta+\gamma)(x^n-x_S)+(\alpha+\beta)(I-A^TA)(x^n-x_S)-\beta(I-A^TA)(x^{n+1}-x_S)+(\beta-\gamma)(x^{n+1}-x_S)+\alpha A^Te'.$$

It implies

$$\begin{aligned}
&\|(x^{n+1}-x_S)_{\overline{S^{n+1}}}\|-\|(u^{n+1}-x_S)_{S\setminus S^{n+1}}\|\\
\geq&\|(x^{n+1}-x_S)_{\overline{S^{n+1}}}\|-\|\Big((1-\alpha-\beta+\gamma)(x^n-x_S)+(\alpha+\beta)(I-A^\top A)(x^n-x_S)\\
&-\beta(I-A^\top A)(x^{n-1}-x_S)+(\beta-\gamma)(x^{n-1}-x_S)\Big)_{S\setminus S^{n+1}}\|-\alpha\|\left(A^\top e'\right)_{S\setminus S^{n+1}}\|,
\end{aligned}$$

and

$$\begin{aligned}
&\|(u^{n+1}-x_S)_{S^{n+1}\setminus S}\|\\
\leq&\|\Big((1-\alpha-\beta+\gamma)(x^n-x_S)+(\alpha+\beta)(I-A^\top A)(x^n-x_S)-\beta(I-A^\top A)(x^{n-1}-x_S)\\
&+(\beta-\gamma)(x^{n-1}-x_S)\Big)_{S^{n+1}\setminus S}\|+\alpha\|(A^\top e')_{S^{n+1}\setminus S}\|.
\end{aligned}$$

It then follows from (3.5) that

$$\begin{aligned}
&\|(x^{n+1}-x_S)_{\overline{S^{n+1}}}\|\\
\leq&\|(u^{n+1}-x_S)_{S\setminus S^{n+1}}\|+\|(u^{n+1}-x_s)_{S^{n+1}\setminus S}\|\\
\leq&\sqrt{2}\|(u^{n+1}-x_S)_{S\triangle S^{n+1}}\|\\
\leq&\sqrt{2}\Big[\|((1-\alpha-\beta+\gamma)(x^n-x_S)+(\alpha+\beta)(I-A^\top A)(x^n-x_S)-\beta(I-A^\top A)(x^{n-1}-x_S)\\
&+(\beta-\gamma)(x^{n-1}-x_S))_{S^{n+1}\triangle S}\|+\alpha\|(A^\top e')_{S^{n+1}\triangle S}\|\Big]\\
\leq&\sqrt{2}\Big[(|1-\alpha-\beta+\gamma|+(\alpha+\beta)\delta_{3s})\|x^n-x_S\|+(\beta\delta_{3s}+|\beta-\gamma|)\|x^{n-1}-x_S\|+\alpha\sqrt{1+\delta_{2s}}\|e'\|\Big],
\end{aligned}\tag{3.6}$$

where in the last step we use the facts that $|\mathrm{supp}(x^n-x_S)\cup(S\triangle S^{n+1})|\leq 3s$ and $|\mathrm{supp}(x^{n-1}-x_S)\cup(S\triangle S^{n+1})|\leq 3s$ by Lemma 3.1. Substituting (3.6) into (3.3) yields

$$\begin{aligned}
&\|x^{n+1}-x_S\|\\
\leq&\frac{1}{\sqrt{1-\delta_{2s}}}\|(x^{n+1}-x_S)_{\overline{S^{n+1}}}\|+\frac{\sqrt{1+\delta_s}}{\sqrt{1-\delta_{2s}}}\|e'\|\\
\leq&\sqrt{\frac{2}{1-\delta_{2s}^2}}\Big[(|1-\alpha-\beta+\gamma|+(\alpha+\beta)\delta_{3s})\|x^n-x_S\|+(\beta\delta_{3s}+|\beta-\gamma|)\|x^{n-1}-x_S\|\\
&+\alpha\sqrt{1+\delta_{2s}}\|e'\|\Big]+\frac{\sqrt{1+\delta_s}}{\sqrt{1-\delta_{2s}}}\|e'\|\\
=&\sqrt{\frac{2}{1-\delta_{2s}^2}}\Big[(|1-\alpha-\beta+\gamma|+(\alpha+\beta)\delta_{3s})\|x^n-x_S\|+(\beta\delta_{3s}+|\beta-\gamma|)\|x^{n-1}-x_S\|\Big]\\
&+\left(\frac{\alpha\sqrt{2(1+\delta_{2s})}}{\sqrt{1-\delta_{2s}^2}}+\frac{\sqrt{1+\delta_s}}{\sqrt{1-\delta_{2s}}}\right)\|e'\|\\
=&\widehat{b}\|x^n-x_S\|+\widehat{c}\|x^{n-1}-x_S\|+(1-\theta)\widehat{\tau}\|e'\|,
\end{aligned}\tag{3.7}$$

where $\widehat{b}, \widehat{c}$, and $\widehat{\tau}$ are given in Theorem 3.1.

To establish the convergence of iterative sequence $\{x^n\}$, according to Lemma 3.2, it needs to ensure that $\widehat{b} + \widehat{c} < 1$. This can be achieved by restricting the range of parameters α, β, γ. Since $\delta_{2s} \leq \delta_{3s} < 1/\sqrt{3} \approx 0.577$ by assumption, then $\sqrt{2}\delta_{3s} < \sqrt{1-\delta_{3s}^2} \leq \sqrt{1-\delta_{2s}^2} = \sqrt{2}/\widehat{\eta}$, which implies $0 < 1/\widehat{\eta} - \delta_{3s}$ and $0 < \frac{\frac{1}{\widehat{\eta}}-\delta_{3s}}{1+\delta_{3s}} < \frac{\frac{1}{\widehat{\eta}}-\delta_{3s}}{2\delta_{3s}}$. This ensures that two ranges of β in (3.1) are well-defined. Furthermore, from the conditions $(1/\hat{\eta} - \delta_{3s})/(1 + \delta_{3s}) \leq \beta < (1/\hat{\eta} - \delta_{3s})/(2\delta_{3s})$ or $0 \leq \beta < (1/\hat{\eta} - \delta_{3s})/(2\delta_{3s})$, we know that parameters α, γ given in (3.1) are also well-defined. Let us now discuss the following two cases based on the ranges of α and γ specified in (3.1).

Case (i). If (α, γ) satisfies case (I) in (3.1), then $\beta > \gamma$. We further consider the following two subcases.
(i)-1. $\alpha \leq 1 - \beta + \gamma$. In this case

$$\begin{aligned}
\widehat{b} &= \widehat{\eta}[|1-\alpha-\beta+\gamma| + (\alpha+\beta)\delta_{3s}] = \widehat{\eta}[1-\alpha-\beta+\gamma+(\alpha+\beta)\delta_{3s}] \\
&= \widehat{\eta}[1-(1-\delta_{3s})\alpha-(1-\delta_{3s})\beta+\gamma] < \widehat{\eta}\left[1-(1+2\beta\delta_{3s}-\frac{1}{\widehat{\eta}})-(1-\delta_{3s})\beta+\gamma\right] \\
&= 1-\widehat{\eta}(\beta\delta_{3s}+\beta-\gamma) = 1-\widehat{\eta}(\beta\delta_{3s}+|\beta-\gamma|) = 1-\widehat{c},
\end{aligned}$$

where the inequality comes from the fact that $\alpha > (1 + 2\beta\delta_{3s} - \frac{1}{\widehat{\eta}})/(1 - \delta_{3s})$.
(i)-2. $\alpha > 1 - \beta + \gamma$. In this case

$$\begin{aligned}
\widehat{b} &= \widehat{\eta}[|1-\alpha-\beta+\gamma| + (\alpha+\beta)\delta_{3s}] = \widehat{\eta}[-1+\alpha+\beta-\gamma+(\alpha+\beta)\delta_{3s}] \\
&= \widehat{\eta}[-1+(1+\delta_{3s})\alpha+(1+\delta_{3s})\beta-\gamma] \\
&< \widehat{\eta}\left[-1+\left(\frac{1}{\widehat{\eta}}+1-2(\delta_{3s}+1)\beta+2\gamma\right)+(1+\delta_{3s})\beta-\gamma\right] \\
&= 1-\widehat{\eta}(\beta\delta_{3s}+\beta-\gamma) = 1-\widehat{\eta}(\beta\delta_{3s}+|\beta-\gamma|) = 1-\widehat{c},
\end{aligned}$$

where the inequality comes from the fact that $\alpha < (\frac{1}{\widehat{\eta}} + 1 - 2(\delta_{3s} + 1)\beta + 2\gamma)/(1 + \delta_{3s})$.
Case (ii). If (α, γ) satisfies case (II) in (3.1), then $\beta < \gamma$. We further consider the following two subcases.
(ii)-1. $\alpha \leq 1 - \beta + \gamma$. In this case

$$\begin{aligned}
\widehat{b} &= \widehat{\eta}[|1-\alpha-\beta+\gamma| + (\alpha+\beta)\delta_{3s}] = \widehat{\eta}[1-\alpha-\beta+\gamma+(\alpha+\beta)\delta_{3s}] \\
&= \widehat{\eta}[1-(1-\delta_{3s})\alpha-(1-\delta_{3s})\beta+\gamma] < \widehat{\eta}\left[1-(1+2\beta\delta_{3s}-\frac{1}{\widehat{\eta}}+2\gamma-2\beta)-(1-\delta_{3s})\beta+\gamma\right] \\
&= 1-\widehat{\eta}(\beta\delta_{3s}-\beta+\gamma) = 1-\widehat{\eta}(\beta\delta_{3s}+|\beta-\gamma|) = 1-\widehat{c},
\end{aligned}$$

where the inequality comes from the fact that $\alpha > (1 + 2\beta\delta_{3s} - \frac{1}{\widehat{\eta}} + 2\gamma - 2\beta)/(1 - \delta_{3s})$.
(ii)-2. $\alpha > 1 - \beta + \gamma$. In this case

$$\begin{aligned}
\widehat{b} &= \widehat{\eta}[|1-\alpha-\beta+\gamma| + (\alpha+\beta)\delta_{3s}] = \widehat{\eta}[-1+\alpha+\beta-\gamma+(\alpha+\beta)\delta_{3s}] \\
&= \widehat{\eta}[-1+(1+\delta_{3s})\alpha+(1+\delta_{3s})\beta-\gamma] \\
&< \widehat{\eta}\left[-1+\left(\frac{1}{\widehat{\eta}}+1-2\beta\delta_{3s}\right)+(1+\delta_{3s})\beta-\gamma\right]
\end{aligned}$$

$$= 1 - \widehat{\eta}(\beta\delta_{3s} - \beta + \gamma) = 1 - \widehat{\eta}(\beta\delta_{3s} + |\beta - \gamma|) = 1 - \widehat{c},$$

where the inequality comes from the fact that $\alpha < (\frac{1}{\widehat{\eta}} + 1 - 2\beta\delta_{3s})/(1 + \delta_{3s})$.

In both cases, we have $\widehat{b} + \widehat{c} < 1$. Applying Lemma 3.2 to (3.7) yields the desired (3.2). This completes the proof. □

4. Number of iterations

Inspired by the work in [33], in this section, we focus on deriving an estimate of the number of iterations that exhibits a uniform property. The following lemma shows that the indices of nonzero entries of x can be correctly captured by the support set S^k.

Lemma 4.1. *Let $x \in \mathbb{R}^N$ be s-sparse and let S^{n+1} be the index set produced by AOR-HBHTP. Suppose that $S^{n+1} \cap S^{n+2}$ contains the indices of p largest absolute entries of x. If there exist positive scalars k, q such that*

$$x^*_{p+q} \geq 2\theta^{k-1}\|x^*_{\{p+1,\ldots,s\}}\|, \tag{4.1}$$

where θ is given in Theorem 3.1, then $S^{n+1+k} \cap S^{n+2+k}$ contains the indices of $p + q$ largest absolute entries of x.

Proof. Let $x^* \in \mathbb{R}^N_+$ be a rearrangement of $|x|$, satisfying $x^*_1 \geq x^*_2 \geq \ldots \geq x^*_N \geq 0$. Denote by π the permutation of $\{1, 2, \ldots, N\}$, satisfying $|x_{\pi(j)}| = x^*_j$ for $j \in \{1, 2, \ldots, N\}$. By assumption, we know $\pi(\{1, 2, \ldots, p\}) \subset S^{n+1} \cap S^{n+2}$. It needs us to prove

$$\pi(\{1, 2, \ldots, p + q\}) \subset S^{n+1+k} \cap S^{n+2+k}.$$

We first show that $\pi(\{1, 2, \ldots, p + q\}) \subset S^{n+1+k}$, i.e., $|(u^{n+1+k})_{\pi(j)}|$ for $j \in \{1, \ldots, p + q\}$ are among the s largest values of $|(u^{n+1+k})_i|$ for $i \in \{1, \ldots, N\}$. It is sufficient to prove

$$\min_{j\in\{1,\ldots,p+q\}} |(u^{n+1+k})_{\pi(j)}| \geq \max_{l\in\overline{S}} |(u^{n+1+k})_l|, \tag{4.2}$$

where $\bar{S} := \overline{\mathrm{supp}(x)}$. For the simplicity of notations, let us denote

$$B(n,k) := (1-\alpha-\beta+\gamma)(x^{n+k}-x)+(\alpha+\beta)(I-A^TA)(x^{n+k}-x)+(\beta-\gamma)(x^{n+k-1}-x)-\beta(I-A^TA)(x^{n+k-1}-x).$$

Hence, $u^{n+1+k} = x + B(n, k)$. Thus,

$$\begin{aligned} |(u^{n+1+k})_{\pi(j)}| &= |(x + B(n,k))_{\pi(j)}| \geq |x_{\pi(j)}| - |B(n,k)_{\pi(j)}| \\ &\geq x^*_{p+q} - |B(n,k)_{\pi(j)}|, \quad \forall j \in \{1, 2, \cdots, p + q\}, \end{aligned}$$

and

$$|u^{n+1+k}_l| = |(x + B(n,k))_l| = |x_l + B(n,k)_l| = |B(n,k)_l|, \quad \forall l \in \bar{S}.$$

Therefore, (4.2) can be proven provided that we show

$$x^*_{p+q} \geq |(B)_{\pi(j)}| + |(B)_l|, \quad \forall j \in \{1, \ldots, p + q\} \text{ and } l \in \overline{S}. \tag{4.3}$$

Let

$$q := \frac{-\widehat{b} + \sqrt{\widehat{b}^2 + 4\widehat{c}}}{2} \text{ and } \theta := \widehat{b} + q, \tag{4.4}$$

where $\widehat{b}, \widehat{c}$ are given in Theorem 3.1. Then, $\theta < 1$ by Lemma 3.2 and

$$(\widehat{b} + q)q = \widehat{c}. \tag{4.5}$$

Note that

$$\begin{aligned} \widehat{b}\|x^{n+k} - x\| + \widehat{c}\|x^{n+k-1} - x\| &\leq (\widehat{b} + q)\|x^{n+k} - x\| + (\widehat{b} + q)q\|x^{n+k-1} - x\| \\ &= \theta(\|x^{n+k} - x\| + q\|x^{n+k-1} - x\|). \end{aligned} \tag{4.6}$$

Hence,

$$\begin{aligned} \|x^{n+k} - x\| + q\|x^{n+k-1} - x\| &\leq \hat{b}\|x^{n+k-1} - x\| + \hat{c}\|x^{n+k-2} - x\| + q\|x^{n+k-1} - x\| \\ &= (\hat{b} + q)\|x^{n+k-1} - x\| + \hat{c}\|x^{n+k-2} - x\| \\ &= (\hat{b} + q)\|x^{n+k-1} - x\| + (\hat{b} + q)q\|x^{n+k-2} - x\| \\ &= \theta(\|x^{n+k-1} - x\| + q\|x^{n+k-2} - x\|), \end{aligned}$$

where the first step follows from (3.7), the third step is due to (4.5), and the last step comes from the fact that $\theta = \hat{b} + q$ by definition (4.4). This leads to

$$\|x^{n+k} - x\| + q\|x^{n+k-1} - x\| \leq \theta^{k-2}(\|x^{n+2} - x\| + q\|x^{n+1} - x\|). \tag{4.7}$$

Therefore,

$$\begin{aligned} &|B(n,k)_{\pi(j)}| + |B(n,k)_l| \\ &\leq \sqrt{2\|B(n,k)_{\{\pi(j),l\}}\|^2} \\ &= \sqrt{2}\left\|\Big[(1-\alpha-\beta+\gamma)(x^{n+k}-x) + (\alpha+\beta)(I-A^TA)(x^{n+k}-x) + (\beta-\gamma)(x^{n+k-1}-x)\right. \\ &\qquad \left.-\beta(I-A^TA)(x^{n+k-1}-x)\Big]_{\{\pi(j),l\}}\right\| \\ &\leq \sqrt{2}\Big[|1-\alpha-\beta+\gamma|\|(x^{n+k}-x)_{\{\pi(j),l\}}\| + (\alpha+\beta)\left\|[(I-A^TA)(x^{n+k}-x)]_{\{\pi(j),l\}}\right\| \\ &\qquad + |\beta-\gamma|\|(x^{n+k-1}-x)_{\{\pi(j),l\}}\| + \beta\left\|[(I-A^TA)(x^{n+k-1}-x)]_{\{\pi(j),l\}}\right\|\Big] \\ &\leq \sqrt{2}\Big[|1-\alpha-\beta+\gamma|\|(x^{n+k}-x)\| + (\alpha+\beta)\delta_{2s+1}\|x^{n+k}-x\| + |\beta-\gamma|\|x^{n+k-1}-x\| \\ &\qquad + \beta\delta_{2s+1}\|x^{n+k-1}-x\|\Big] \\ &= \sqrt{2}\Big[\Big(|1-\alpha-\beta+\gamma| + (\alpha+\beta)\delta_{2s+1}\Big)\|(x^{n+k}-x)\| + \Big(|\beta-\gamma| + \beta\delta_{2s+1}\Big)\|x^{n+k-1}-x\|\Big] \\ &\leq \sqrt{2}\Big[\Big(|1-\alpha-\beta+\gamma| + (\alpha+\beta)\delta_{3s}\Big)\|(x^{n+k}-x)\| + \Big(|\beta-\gamma| + \beta\delta_{3s}\Big)\|x^{n+k-1}-x\|\Big] \\ &= \frac{\sqrt{2}}{\widehat{\eta}}\Big(\widehat{b}\|x^{n+k}-x\| + \widehat{c}\|x^{n+k-1}-x\|\Big) \end{aligned}$$

$$\leq \frac{\sqrt{2}}{\widehat{\eta}}\theta\Big(\|x^{n+k}-x\|+q\|x^{n+k-1}-x\|\Big)$$
$$\leq \frac{\sqrt{2}}{\widehat{\eta}}\theta^{k-1}\Big(\|x^{n+2}-x\|+q\|x^{n+1}-x\|\Big), \tag{4.8}$$

where the fourth step comes from Lemma 3.1 since $|\mathrm{supp}(x^{n+k}-x)\cup\{\pi(j),l\}|\leq 2s+1$ and $|\mathrm{supp}(x^{n+k-1}-x)\cup\{\pi(j),l\}|\leq 2s+1$, the seventh step follows from the definition of $\hat{b},\hat{c},\hat{\eta}$ given in Theorem 3.1, the eighth step is due to (4.6), and the last step comes from (4.7).

According to (3.3), we have

$$\frac{\sqrt{2}}{\widehat{\eta}}\theta^{k-1}(\|x^{n+2}-x_S\|+q\|x^{n+1}-x_S\|)\leq \frac{\sqrt{2}}{\widehat{\eta}}\theta^{k-1}\left(\frac{1}{\sqrt{1-\delta_{2s}^2}}\|x_{\overline{S^{n+2}}}\|+\frac{q}{\sqrt{1-\delta_{2s}^2}}\|x_{\overline{S^{n+1}}}\|\right)$$
$$=\theta^{k-1}(\|x_{\overline{S^{n+2}}}\|+q\|x_{\overline{S^{n+1}}}\|), \tag{4.9}$$

where the last step is due to $\widehat{\eta}:=\sqrt{2/(1-\delta_{3s}^2)}$ by Theorem 3.1. Because $\pi\{1,2,...,p\}\subset S^{n+1}\cap S^{n+2}$, then $\overline{S^{n+1}}\subset\overline{\pi\{1,2,...,p\}}$ and $\overline{S^{n+2}}\subset\overline{\pi\{1,2,...,p\}}$. Hence, it follows from (4.8) and (4.9) that

$$|B(n,k)_{\pi(j)}|+|B(n,k)_l|\leq\theta^{k-1}(\|x_{\overline{S^{n+2}}}\|+q\|x_{\overline{S^{n+1}}}\|)\leq\theta^{k-1}(\|x_{\overline{\pi\{1,2,...,p\}}}\|+q\|x_{\overline{\pi\{1,2,...,p\}}}\|)$$
$$=\theta^{k-1}(1+q)\|x^*_{\{p+1,...,s\}}\|\leq 2\theta^{k-1}\|x^*_{\{p+1,...,s\}}\|\leq x^*_{p+q},$$

where the fourth step follows from the fact that $q=\theta-\widehat{b}<\theta<1$ by (4.4) and Lemma 3.2, and the last step is due to (4.1). Hence, (4.3) holds, which further implies (4.2). This proves $\pi(\{1,2,...,p+q\})\subset S^{n+1+k}$. Note that the assumption (4.1) still holds if k is replaced by $k+1$. By following a similar argument, we can obtain $\pi(\{1,2,...,p+q\})\subset S^{n+1+(k+1)}=S^{n+k+2}$. Putting these two results together yields $\pi(\{1,2,...,p+q\})\subset S^{n+1+k}\cap S^{n+2+k}$. This completes the proof. □

The following result provides an estimate of the number of iterative steps. It is important to note that the established upper bound is uniform in the sense that it is proportional to the sparsity level, and independent of the specific shape of the unknown signal to be recovered.

Theorem 4.1. *Let $c_0>2$ be a given positive integer, and define $t_0:=\frac{1}{2}\left(\frac{1}{16}\right)^{\frac{1}{c_0-1}}$ and $\widehat{\eta}:=\sqrt{\frac{2}{1-\delta_{3s}^2}}$. Suppose that the RIC of the matrix $A\in\mathbb{R}^{m\times N}$ satisfies*

$$\delta_{3s}\leq\sqrt{\frac{\left(\frac{1}{16}\right)^{\frac{2}{c_0-1}}}{8+\left(\frac{1}{16}\right)^{\frac{2}{c_0-1}}}}. \tag{4.10}$$

Furthermore, choose parameters α,β,γ such that one of the following two cases

$$\begin{cases}(I): & \frac{t_0}{\widehat{\eta}}-\delta_{3s}\leq\beta<\frac{\frac{t_0}{\widehat{\eta}}-\delta_{3s}}{2\delta_{3s}}, \quad \frac{\delta_{3s}-\frac{t_0}{\widehat{\eta}}+(1+\delta_{3s})\beta}{1-\delta_{3s}}<\gamma<\beta, \quad \frac{1+2\beta\delta_{3s}-\frac{t_0}{\widehat{\eta}}}{1-\delta_{3s}}<\alpha<\frac{\frac{t_0}{\widehat{\eta}}+1-2(\delta_{3s}+1)\beta+2\gamma}{1+\delta_{3s}},\\ (II): & 0\leq\beta<\frac{\frac{t_0}{\widehat{\eta}}-\delta_{3s}}{2\delta_{3s}}, \quad \beta<\gamma<\frac{-\delta_{3s}+\frac{t_0}{\widehat{\eta}}+(1-\delta_{3s})\beta}{1+\delta_{3s}}, \quad \frac{1+2\beta\delta_{3s}-\frac{t_0}{\widehat{\eta}}+2\gamma-2\beta}{1-\delta_{3s}}<\alpha<\frac{\frac{t_0}{\widehat{\eta}}+1-2\beta\delta_{3s}}{1+\delta_{3s}}.\end{cases} \tag{4.11}$$

Then, every s-sparse vector $x\in\mathbb{R}^N$ can be recovered from the measurement $y=Ax$ via a number $\bar{n}$ of iterations of AOR-HBHTP with

$$\bar{n}\leq c_0 s.$$

Proof. According to the ranges of δ_{3s} and α, β, γ given in (4.10) and (4.11), we can obtain

$$\hat{b} + \hat{c} < \frac{1}{2} \cdot \left(\frac{1}{16}\right)^{\frac{1}{c_0-1}} = t_0 < \frac{1}{2}, \tag{4.12}$$

by following a similar argument as in Theorem 3.1. Consequently the iterative sequence $\{x^n\}$ generated by AOR-HBHTP converges. Recalling the definition of θ given in Lemma 3.2, it follows from (4.12) that

$$\theta^2 = \left(\frac{\hat{b} + \sqrt{\hat{b}^2 + 4\hat{c}}}{2}\right)^2 \leq \hat{b}^2 + 2\hat{c} \leq 2(\hat{b} + \hat{c}) < \left(\frac{1}{16}\right)^{\frac{1}{c_0-1}}. \tag{4.13}$$

As shown in Lemma 4.1, let π be the permutation of $\{1, 2, ..., N\}$ satisfying $|x_{\pi(j)}| = x_j^*$ for all $j \in \{1, 2, ..., N\}$. First choose $\zeta \in (0, 1)$ and let $q_0 := 0$. Consider a special partition of the support set supp(x) as follows: There exists a positive integer $r \leq s$ such that $Q_1 \cup Q_2 \cup \cdots \cup Q_r = \text{supp}(x)$, where

$$Q_i := \pi(\{q_{i-1} + 1, ..., q_i\}),$$

and q_i denotes the maximum index satisfying $q_i \geq q_{i-1} + 1$ and $x_{q_i}^* \geq \sqrt{\zeta} x_{q_{i-1}+1}^*$. For $i = 1, \cdots, r$, define

$$k_i := 1 + \left\lfloor \frac{\ln\left[\frac{4}{\zeta}(|Q_i| + \zeta|Q_{i+1}| + \cdots + \zeta^{r-i}|Q_r|)\right]}{\ln \frac{1}{\theta^2}} \right\rfloor. \tag{4.14}$$

Let $Q_0 := \phi$ and $k_0 := 0$. We now claim that

$$Q_0 \cup Q_1 \cup Q_2 \cup ... \cup Q_i \subset S^{k_0+k_1+...+k_i} \cap S^{k_0+k_1+...+k_i+1}, \quad \forall i = 0, 1, 2, ..., r. \tag{4.15}$$

Indeed, as if i=0, (4.15) holds trivially due to $Q_0 = \emptyset$. Now assume that (4.15) holds as $i = 0, 1, \cdots, r-1$. Recall that $x_{q_i+1}^* < \sqrt{\zeta} x_{q_{i-1}+1}^*$. Hence,

$$\begin{aligned} \|x_{Q_i}\|^2 + \|x_{Q_{i+1}}\|^2 ... + \|x_{Q_r}\|^2 \leq & (x_{q_{i-1}+1}^*)^2|Q_i| + (x_{q_i+1}^*)^2|Q_{i+1}| + ... + (x_{q_{r-1}+1}^*)^2|Q_r| \\ \leq & (x_{q_{i-1}+1}^*)^2[|Q_i| + \zeta|Q_{i+1}| + ... + \zeta^{r-i}|Q_r|]. \end{aligned} \tag{4.16}$$

Combining (4.14) and (4.16) together yields

$$\begin{aligned} 4\theta^{2(k_i-1)}(\|x_{Q_i}\|^2 + \|x_{Q_{i+1}}\|^2 ... + \|x_{Q_r}\|^2) \leq & 4\theta^{2(k_i-1)}(x_{q_{i-1}+1}^*)^2\Big[|Q_i| + \zeta|Q_{i+1}| + ... + \zeta^{r-i}|Q_r|\Big] \\ = & \frac{4}{\zeta}\theta^{2(k_i-1)}\Big[|Q_i| + \zeta|Q_{i+1}| + ... + \zeta^{r-i}|Q_r|\Big](\sqrt{\zeta} x_{q_{i-1}+1}^*)^2 \\ \leq & (\sqrt{\zeta} x_{q_{i-1}+1}^*)^2 \leq (x_{q_i}^*)^2. \end{aligned}$$

Therefore,

$$x_{q_i}^* \geq 2\theta^{k_i-1}\sqrt{\|x_{Q_i}\|^2 + \|x_{Q_{i+1}}\|^2 ... + \|x_{Q_r}\|^2} = 2\theta^{k_i-1}\|(x_{q_{i-1}+1}, \cdots, x_{q_r})\|.$$

Thus, by Lemma 4.1, we have

$$\pi\{1, 2, \cdots, q_i\} \subset S^{k_0+k_1+...+k_i} \cap S^{k_0+k_1+...+k_i+1}.$$

Since $\pi\{1, 2, \cdots, q_i\} = Q_0 \cup Q_1 \cup ... \cup Q_i$, then $Q_0 \cup Q_1 \cup ... \cup Q_i \subset S^{k_0+k_1+...+k_i} \cap S^{k_0+k_1+...+k_i+1}$. Hence, (4.15) holds true.

Note that the support set $\pi(\{1, ..., s\})$ of x can be recovered in a number of iterations at most

$$\begin{aligned}\bar{n} = \sum_{i=1}^{r} k_i &\leq \sum_{i=1}^{r} \left[1 + \log_{\frac{1}{\theta^2}} \left(\frac{4}{\zeta}\Big[|Q_i| + \zeta|Q_{i+1}| + ... + \zeta^{r-i}|Q_r|\Big]\right)\right] \\ &= r + \sum_{i=1}^{r} \log_{\frac{1}{\theta^2}} \left(\frac{4}{\zeta}\Big[|Q_i| + \zeta|Q_{i+1}| + ... + \zeta^{r-i}|Q_r|\Big]\right).\end{aligned}$$

Hence,

$$\begin{aligned}\frac{\bar{n}-r}{r} \ln\left(\frac{1}{\theta^2}\right) &\leq \sum_{i=1}^{r} \frac{1}{r} \ln\left(\frac{4}{\zeta}\Big[|Q_i| + \zeta|Q_{i+1}| + ... + \zeta^{r-i}|Q_r|\Big]\right) \\ &\leq \ln\left(\sum_{i=1}^{r} \frac{4}{r\zeta}\Big[|Q_i| + \zeta|Q_{i+1}| + ... + \zeta^{r-i}|Q_r|\Big]\right) \\ &= \ln\left(\frac{4}{r\zeta}\Big[|Q_1| + (1+\zeta)|Q_2| + ... + (1+\zeta+...+\zeta^{r-1})|Q_r|\Big]\right) \\ &\leq \ln\left(\frac{4}{r\zeta(1-\zeta)}\Big[|Q_1| + |Q_2| + ... + |Q_r|\Big]\right) \\ &\leq \ln\left(\frac{4s}{r\zeta(1-\zeta)}\right), \end{aligned} \tag{4.17}$$

where the second step follows from the concavity of the logarithm and the last step comes from the fact that $|Q_1| + |Q_2| + ... + |Q_r| = |Q_1 \cup Q_2 \cup \cdots \cup Q_r| = |\text{supp}(x)| \leq s$. Note that $\ln\left(\frac{4s}{r\zeta(1-\zeta)}\right)$ for $\zeta \in (0, 1)$ attains its minimum value if and only if $\zeta = \frac{1}{2}$, and in this case the right hand of (4.17) is $\ln(16s/r)$. Hence,

$$\frac{\bar{n}-r}{s} \ln\left(\frac{1}{\theta^2}\right) = \frac{r}{s} \cdot \frac{\bar{n}-r}{r} \ln\left(\frac{1}{\theta^2}\right) \leq \frac{r}{s} \ln\left(\frac{16s}{r}\right) = \frac{\ln(16s/r)}{s/r} \leq \ln(16),$$

where the last step is due to $\ln(16t)/t \leq \ln(16)$ for $t \geq 1$. Thus,

$$\bar{n} \leq r + \frac{\ln(16)}{\ln\left(\frac{1}{\theta^2}\right)} s \leq \left(1 + \frac{\ln(16)}{\ln\left(\frac{1}{\theta^2}\right)}\right) s \leq c_0 s,$$

where the second step is due to $r \leq s$ and the last step follows from the fact $\ln(16)/\ln(1/\theta^2) < c_0 - 1$ by (4.13). This completes the proof. □

5. Numerical experiments

All numerical experiments are performed on a PC equipped with a 13th Generation Intel(R) Core(TM) i9-13900HX processor operating at 2.20 GHz and 16.0 GB of RAM..

For problem (1.2), the measurement matrix $A \in \mathbb{R}^{m \times N}$ is selected as a Gaussian matrix. Each entry of A is independently and identically distributed (i.i.d.), following a normal distribution $\mathcal{N}(0, m^{-1})$.

This choice ensures that the RIP holds with high probability provided that m is sufficiently enough; see [2, Chapter 9]. The sparse vector $x \in \mathbb{R}^N$ is generated randomly. The positions of the non-zero elements in x are determined by a uniform distribution, while the values at these positions are i.i.d. and follow the normal distribution $\mathcal{N}(0, 1)$. The observation vector y is computed as $y = Ax$ in noiseless environments, and $y = Ax + \lambda e$ in noisy environments, where $\lambda > 0$ represents the noise level and $e \in \mathbb{R}^m$ is a noise vector with entries i.i.d. from $\mathcal{N}(0, m^{-1})$.

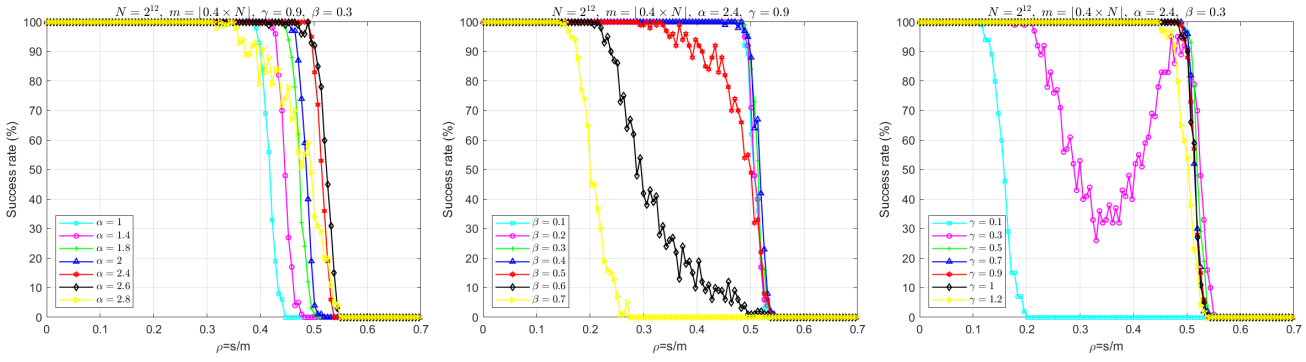
We evaluate the performance of our proposed algorithm, AOR-HBHTP, against six widely recognized algorithms: IHT, HTP, SP, CoSaMP, HBHTP, and EDOMP in terms of reconstruction accuracy and computational efficiency. Experimental parameters are selected as follows: For IHT and HTP, the parameter α is fixed to 1; for HBHTP, $\alpha = 1.7$ and $\gamma = 0.7$ as suggested in [29]; for AOR-HBHTP, $\alpha = 2.4$, $\gamma = 0.9$, and $\beta = 0.3$. The maximum number of iterations for HTP, SP, CoSaMP, HBHTP, and AOR-HBHTP is set to 50, while for EDOMP it is set to s. To address the issue of lacking of pursuit steps, the maximum number of iterations for IHT is increased to 100. Initialization for HBHTP and AOR-HBHTP begins with $x^1 = x^0 = 0$, while other iterative algorithms start from $x^0 = 0$. The recovery of an algorithm is regarded *successful* only if the iteration point x^{n+1} generated by algorithm meets the criterion

$$\|x^{n+1} - \bar{x}\|/\|\bar{x}\| \leq 10^{-3}, \tag{5.1}$$

where $\bar{x}$ represents the true solution.

5.1. Choices of parameters α, β, γ

For the AOR-HBHTP algorithm, the parameters are set as $\alpha = 2.4$, $\beta = 0.3$, and $\gamma = 0.9$. These values are determined through systematic evaluation under specific measurement matrix dimensions $N = 2^{12}$ and $m = \lfloor 0.4N \rfloor$. The size of the measurement matrix $A \in \mathbb{R}^{m \times N}$ directly affects parameter selection, and different testing environments may lead to different parameter values. Figure 1 illustrates the impact of individual parameters (α, β, γ) on the recovery performance of AOR-HBHTP.



(a) Effect of α **(b)** Effect of β **(c)** Effect of γ

Figure 1. Success rates of AOR-HBHTP with different parameters.

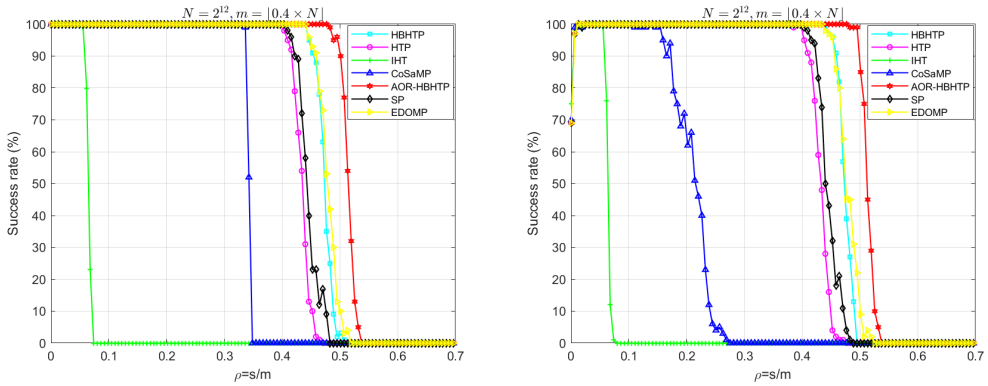
5.2. Comparison of numerical performance

In this section, all tests are conducted in the setting of $N = 2^{12}$ and $m = \lfloor 0.4N \rfloor$. Let $\rho := s/m$ represent the oversampling rate, where s ranges from 1 to 1291 with a step size of 10. For every fixed

sparsity level ρ, 100 pairs $(A, \bar{x})$ are randomly generated using the aforementioned method. The noise level parameter λ is set to 0.02.

5.2.1. Recovery success rate

The recovery success rates of seven algorithms are reported in Figure 2, where Figures 2 (a) and (b) illustrate the results in noiseless and noisy settings. In these figures, the horizontal axis represents the oversampling rate ρ, while the vertical axis denotes the success rate. For each value of ρ, the algorithms are applied to solve 100 randomly selected problems, and the number of successful recoveries is recorded. Figure 2 shows that AOR-HBHTP outperforms the other six algorithms, regardless of the presence of noise. This suggests that AOR-HBHTP exhibits a certain degree of robustness. In contrast, the algorithm CoSaMP is highly sensitive to noise; for example, the success rate remains at 100% when $\rho = 0.3$ under accurate measurements; however, the success rate drops to 0 at the same $\rho = 0.3$ under inaccurate measurements. In summary, as the oversampling rate ρ increases, AOR-HBHTP maintains a high accuracy recovery accuracy even when other algorithms fail.



(a) Accurate measurement (b) Inaccurate measurement

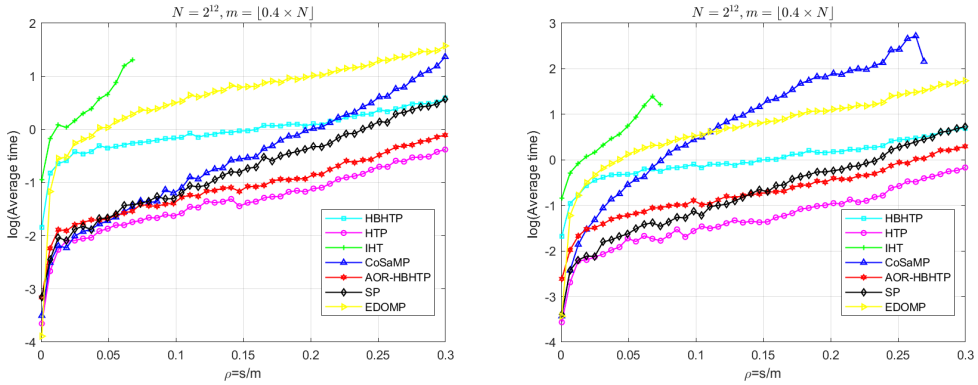
Figure 2. Comparison of success rates.

5.2.2. Average runtime

We focus our attention on comparing the average CPU time of these seven algorithms for signal recovery problems under both accurate measurements and inaccurate measurements. Runtime is measured only for algorithms exhibiting non-zero recovery success rates. Based on the observations from Subsection 5.2.1, numerical results indicate that the recovery success rate of all algorithms except IHT remains at 100% when ρ is less than 0.3, so we primarily consider the case where the oversampling rate ρ is restricted to the interval $(0, 0.3)$. This enables us to record the average CPU time for experiments where all algorithms can successfully recover the signals.

Figure 3 illustrates that CoSaMP consumes more CPU time under noisy conditions than in the noiseless case. The average CPU time spent by all algorithms gradually increases as the oversampling rate ρ rises. Excluding IHT and CoSaMP, among the remaining five algorithms, EDOMP achieves the longest runtime. As the value of ρ increases, our proposed algorithm AOR-HBHTP achieves the second-shortest runtime, surpassed only by HTP. These results demonstrate that AOR-HBHTP exhibits some advantages in terms of sparse signal recovery time as ρ increases. This highlights its efficiency

and robustness in handling higher oversampling rates compared to other algorithms.



(a) Accurate measurement (b) Inaccurate measurement

Figure 3. Comparison of runtime.

5.3. Convergence behavior

We numerically investigate the convergence properties of the algorithms. For parameter values $N = 2^{12}$, $m = \lfloor 0.4N \rfloor$, and $s = \lfloor 0.4m \rfloor$, we record the relative errors between recovered signals and real signals within 5 seconds. IHT and CoSaMP are excluded from Figure 4 because these two algorithms fail to converge in this case.

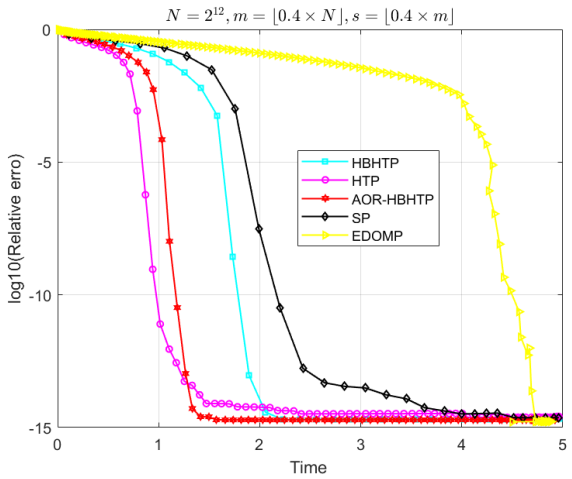


Figure 4. Relative errors.

Figure 4 demonstrates that AOR-HBHTP achieves relatively smaller errors more rapidly than all competing algorithms, confirming its enhanced convergence properties. While other algorithms eventually attain acceptable relative errors, they require longer runtime.

5.4. Phase transition curve and algorithm selection map

Phase Transition Curve (PTC) and Algorithm Selection Map (ASM) are effective tools for demonstrating the recovery capabilities of algorithms; see [34, 35] for more details. In the previous

Section 5.2, we compare the recovery success rates and CPU times of algorithms for a fixed matrix size $m = \lfloor 0.4N \rfloor, N = 2^{12}$. In this section, we introduce a variable δ to control the number of rows m in order to investigate the performance of these algorithms under varying conditions. Here, the number of columns is fixed at $N = 2^{12}$, while the number of rows m varies by setting $m = \lfloor \delta N \rfloor$, where $\delta = \{0.2, 0.25, 0.3, \ldots, 0.8\}$.

For each value of m, we generate 50 groups of spare signals with $s = \lfloor \rho m \rfloor$, where ρ ranges from 0.02 to 1 with a step size of 0.02. The numerical results reflect the recovery ability of algorithms at different sampling rates and sparsity levels. This helps us identify the strengths and limitations of each algorithm across various scenarios.

5.4.1. Phase transition curve

The empirical Phase Transition Curve (PTC) presented in this section is a logistic regression curve described based on a 50% success rate of the algorithms. The PTC of an algorithm divides the (δ, ρ)-plane into two regions: Success and failure (see Figure 5). The area below the curve corresponds to successful recovery, while the area above represents the opposite. For each value of δ, the 'glmfit' function is used to generate the logistic regression curve based on the success rates at different sparsity degrees s. The recovery phase transition curve is drawn by applying a binary method to generate an interval $[s_{\min}, s_{\max}]$, following the same approach as in [29] except for substituting their logistic regression approach with our 'glmfit' function.

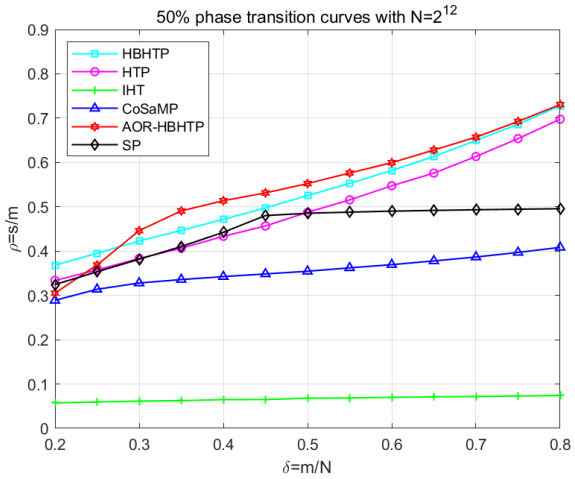


Figure 5. Phase transition curve.

Recall that the EDOMP algorithm requires two hard thresholding and two orthogonal projections per iteration, with its maximum number of iterations set to the sparsity level s. In comparison, other algorithms typically involve only one hard thresholding and one orthogonal projection. As shown in Figures 2 and 3, while EDOMP achieves higher success rates, its computational time increases significantly with increasing $\delta = m/N$ and $\rho = s/m$, especially with increasing sparsity level s. Hence, six algorithms excluding EDOMP are considered when plotting PTC.

The comparison of PTCs for these six algorithms is shown in Figure 5. It can be observed that when $\delta \leq 0.25$, SP and HTP exhibit similar recovery phase transition curves, while HBHTP achieves the highest curve. However, when $\delta \geq 0.3$, AOR-HBHTP attains the highest PTC, indicating that it

outperforms the other five algorithms, particularly in scenarios with larger values of δ. In summary, the comparison of PTCs provides a clear visualization of the algorithms' recovery performance across different undersampling rates δ. The superior performance of AOR-HBHTP at higher δ values highlights its effectiveness and robustness, making it a preferred choice for sparse signal recovery in certain scenarios.

5.4.2. Algorithm selection map

The Algorithm Selection Map (ASM) serves as a tool to determine the algorithm that consumes the least average CPU time during the process of recovering sparse signals. To plot an ASM, we test 50 random samples for each δ value using all algorithms across the phase space grid (δ, ρ), where $\rho := j/50$ and $j = 1, 2, ..., 50$. Testing continues until the recovery success rate drops below 80%. The algorithm that takes the least average CPU runtime is then marked on the map.

Figure 6 shows that for relatively large values of ρ, AOR-HBHTP is the fastest algorithm within the range $0.3 \leq \delta \leq 0.6$, and HBHTP performs better at $0.2 \leq \delta \leq 0.25$, while EDOMP delivers outstanding performance at $0.65 \leq \delta \leq 0.8$. Moreover, HTP emerges as the fastest algorithm for cases with relatively small ρ. The ASM provides a comprehensive and intuitive overview of the fastest algorithms under different combinations of undersampling rates δ and oversampling rates ρ. By focusing on CPU runtime, it offers practical insights for selecting the most efficient algorithm tailored to specific problem parameters.

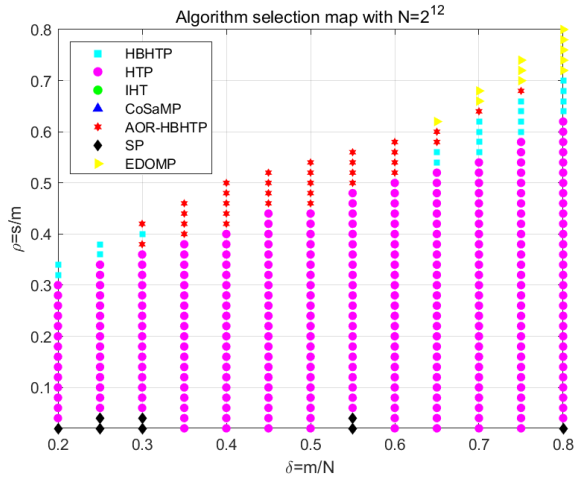
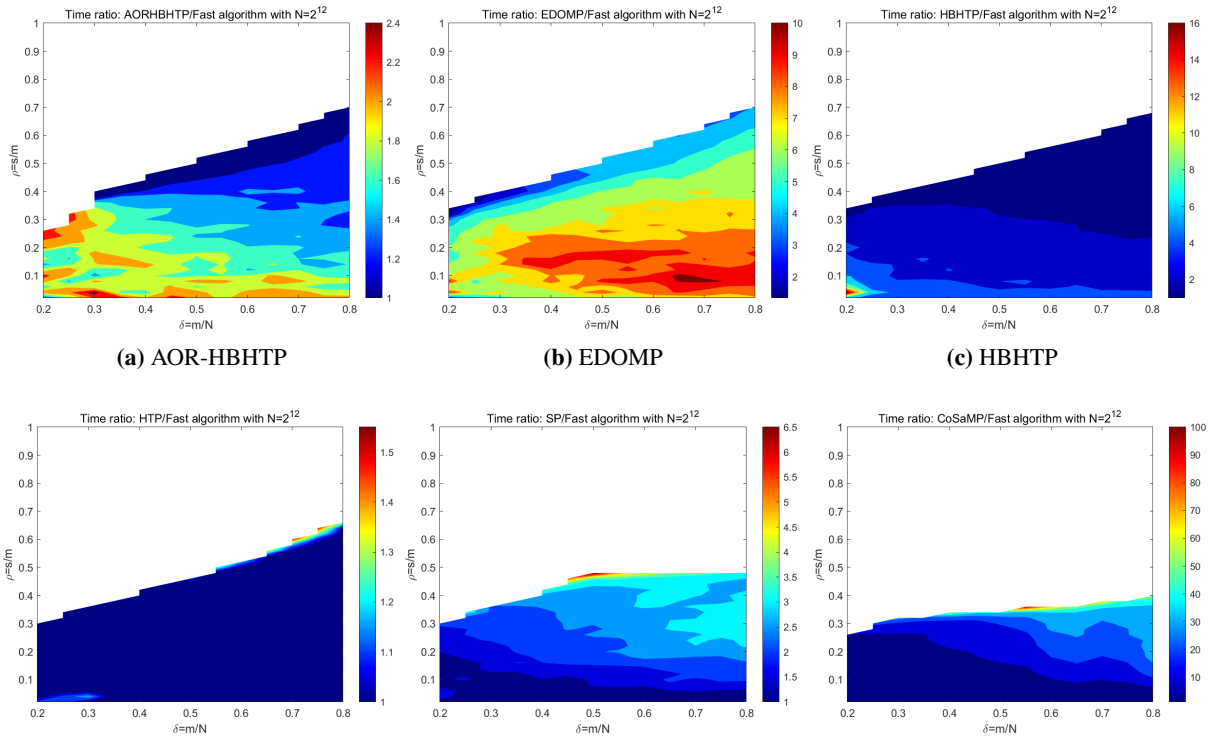


Figure 6. Algorithm selection map.

After identifying the fastest algorithm, the ratio of the average CPU time of algorithms AOR-HBHTP, EDOMP, HBHTP, HTP, SP, and CoSaMP relative to that of the fastest algorithm is displayed in Figure 7(a)–(f), respectively.

Figures 7(a)–(c) indicate that for a fixed δ, as the value of ρ increases, the ratios of the algorithms relative to the fastest one decrease. Specifically, the ratio of AOR-HBHTP is less than 2.2 in most regions, whereas EDOMP and HBHTP are bounded by ratios of 8 and 6. By contrast, Figures 7(d)–(f) show that as the value of ρ increases, the ratios for these algorithms increase. This phenomenon reveals that AOR-HBHTP might work better when the sparsity level s is relatively high. Experimental results show that AOR-HBHTP and HTP exhibit comparable computational efficiency, since their runtime

ratio does not exceed 2.5 in most regions; however, HBHTP and SP typically consume at least four times the minimum CPU time, while CoSaMP exhibit a significant tenfold increase in runtime.



(a) AOR-HBHTP **(b)** EDOMP **(c)** HBHTP

(d) HTP **(e)** SP **(f)** CoSaMP

Figure 7. The ratios of average time relative to the fastest algorithm.

5.5. Applications

To further validate the performance of the proposed algorithms on real-world signals, we employ a seismic signal named `seismic.asc` from the PROMAX seismic data processing system developed by Landmark Graphics Corporation, which can be loaded from https://apptestermc.github.io/Wavelab_850/index.html. The downloaded signal, denoted as the original signal $X \in \mathbb{R}^N$, is a one-dimensional vector of length $N = 1024$. Under this sample length, the number of measurements is set to $m = \lceil \frac{1}{2}N \rceil$, and the sparsity level is chosen as $s = \lceil \frac{4}{9}m \rceil$. Each element of the measurement matrix A is independently drawn from a normal distribution $\mathcal{N}(0, m^{-1})$. To obtain a sparse representation, we employ a 7-level discrete wavelet transform using the `sym7` wavelet basis to yield an orthogonal wavelet transform matrix W.

We first evaluate the sparsity properties of the original signal in the wavelet domain. Specifically, the wavelet coefficient X' is computed by multiplying the original signal X with the previously obtained wavelet transform matrix W, i.e., $X' = WX$. We then select the s coefficients in X' with the largest absolute values to form a sparse signal X'_s, and treat the remaining $n - s$ wavelet coefficients with the smallest absolute values as noise e'. We employ signal recovery algorithms to obtain a sparse coefficient, say $\tilde{X}'_s$, from the measurement $y = AX = AW^{-1}WX = AW^{-1}(X'_s + e') = AW^{-1}X'_s + e$, where $e := AW^{-1}e'$. The final reconstructed signal $\hat{X}$ is obtained through the inverse wavelet transform

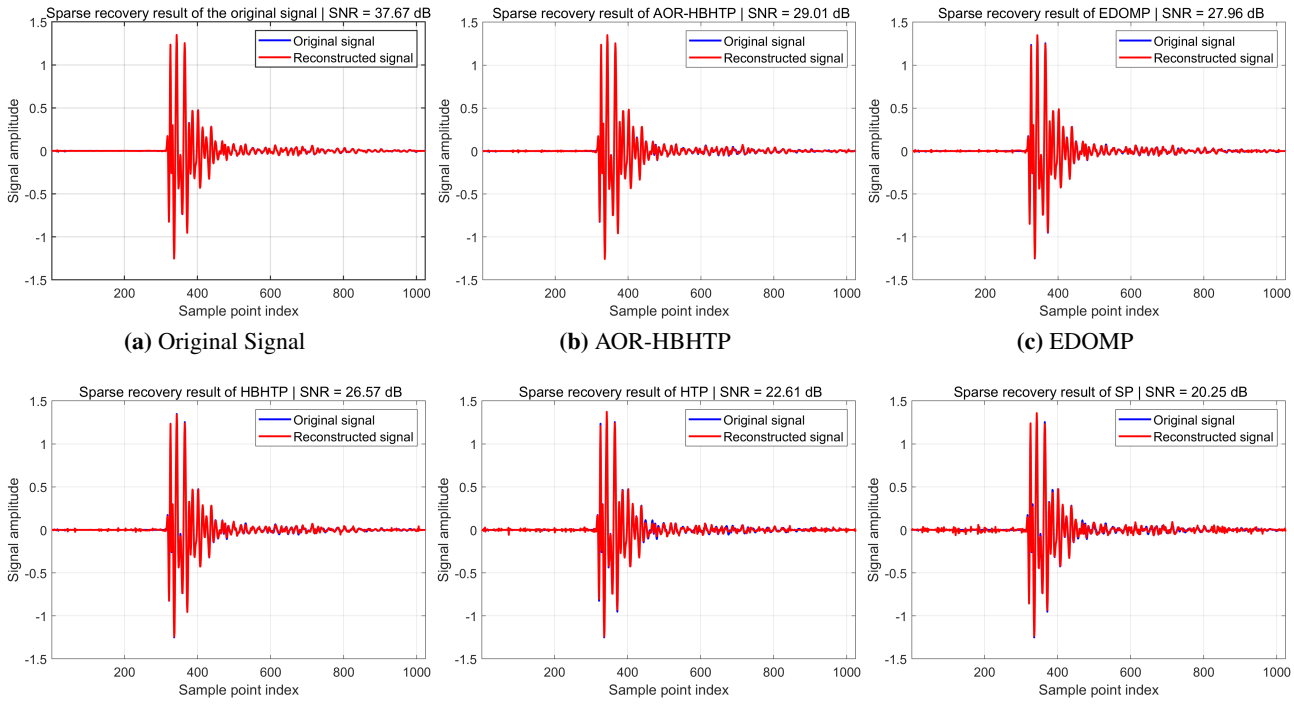
$\hat{X} = W^{-1}\tilde{X}'_s$. The quality of reconstructed signal $\hat{X}$ is quantitatively measured using the Signal-to-Noise Ratio (SNR) defined as

$$\text{SNR} = 10\log_{10}\left(\frac{||X||_2^2}{||X - \hat{X}||_2^2}\right).$$

The higher the SNR value, the better the reconstruction performance.

It should be particularly noted that in real-world scenarios, only the measurement vector y, the measurement matrix A, wavelet transform matrix W, and prespecified sparsity level s are available, whereas the prior knowledge of the true sparse representation of the original signal X is unknown. Therefore, we replace the relative error-based stopping criterion (5.1) with a fixed computational time, specifically requiring that the CPU runtime not exceed 0.5 seconds. The tested algorithms remain numerically convergent trends within this specified time constraint under the given problem settings.

Figure 8 compares reconstruction results of five sparse signal recovery algorithms: HBHTP, AOR-HBHTP, EDOMP, HTP, and SP. Here, CoSaMP and IHT are excluded from our comparison since they fail to converge numerically under this specified experimental conditions. The Signal-to-Noise Ratio (SNR) is adopted as the reconstruction quality metric. Each subplot in Table 2 displays the waveform comparison between original signal (blue) and reconstructed signal (red). Numerical results demonstrate that AOR-HBHTP achieves the highest SNR (29.01 dB), exceeding other algorithms. Particularly, in both the signal peak and flat regions, the reconstructed waveform from AOR-HBHTP is closest to the original signal, which demonstrates its capability in recovering real-world signals.



(a) Original Signal **(b)** AOR-HBHTP **(c)** EDOMP

(d) HBHTP **(e)** HTP **(f)** SP

Figure 8. Comparison of real-world signal recovery.

Table 2. Comparison of SNR.

Algorithms	Original Signal	AOR-HBHTP	EDOMP	HBHTP	HTP	SP
SNR(dB)	37.67	29.01	27.96	26.57	22.61	20.25

6. Conclusions

We have developed a novel hard thresholding algorithm called AOR-HBHTP for solving sparse signal recovery by incorporating the Accelerated Over-Relaxation (AOR) mechanism. By requiring the measurement matrix to satisfy the Restricted Isometry Property (RIP), we establish the convergence analysis of the algorithm and provide an estimate of the required iteration steps. Numerical experiments illustrate that the adding over-relaxation term helps improve the recovery ability in the setting of different levels of noise and sparsity, and make the algorithm more stable by mitigating the issue of excessive oscillations during the iteration process. The AOR mechanism has opened an acceleration strategy distinct from traditional heavy-ball momentum methods. Hence, it is worthwhile to further investigate the extension of this technique to other hard/soft thresholding-type algorithms, aiming to improve recovery capability and robustness in solving compressive sensing problems, and even consider exploring its potential in areas beyond convex optimization problems. Moreover, there are different analysis tools to study sparse signal recovery. Therefore, it is worth to further investigate how to establish the convergence of our proposed algorithm using the mutual coherence property of the measurement matrix.

Author contributions

Xinyu Diao: Methodology, Formal analysis, Algorithm Implementation, Writing-original draft; Zhongfeng Sun: Methodology, Investigation, Writing-review and editing; Jingyong Tang: Verification, Proofreading, Writing-review and editing; Jinchuan Zhou: Supervision, Validation, 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 that they have not used Artificial Intelligence (AI) tools in the creation of this article.

Acknowledgments

The authors are gratefully indebted to the anonymous referees for their valuable suggestions that helped us greatly improve the original presentation of the paper. This work was supported by National Natural Science Foundation of China (12371305) and Shandong Provincial Natural Science Foundation (ZR2023MA020).

Conflict of interest

All authors declare no conflicts of interest in this paper.

References

1. D. Donoho, Compressed sensing, *IEEE Trans. Inform. Theory*, **52** (2006), 1289–1306. https://doi.org/10.1109/TIT.2006.871582
2. S. Foucart, H. Rauhut, An invitation to compressive sensing, In: *A mathematical introduction to compressive sensing*, New York: Springer, 2013. https://doi.org/10.1007/978-0-8176-4948-7_1
3. C. Qiu, X. Hu, AdaCS: adaptive compressive sensing with restricted isometry property-based error-clamping, *IEEE Trans. Pattern Anal.*, **46** (2024), 4702–4719. https://doi.org/10.1109/TPAMI.2024.3357704
4. S. Foucart, Hard thresholding pursuit: an algorithm for compressive sensing, *SIAM J. Numer. Anal.*, **49** (2011), 2543–2563. https://doi.org/10.1137/100806278
5. T. Blumensath, M. Davies, Iterative hard thresholding for compressed sensing, *Appl. Comput. Harmon. Anal.*, **27** (2009), 265–274. https://doi.org/10.1016/j.acha.2009.04.002
6. Y. Feng, A. Taya, Y. Nishiyama, K. Sezaki, J. Liu, Compressive detection of stochastic sparse signals with unknown sparsity degree, *IEEE Signal Proc. Lett.*, **30** (2023), 1482–1486. https://doi.org/10.1109/LSP.2023.3324573
7. W. Xiong, J. Cao, S. Li, Sparse signal recovery with unknown signal sparsity, *EURASIP J. Adv. Signal Process.*, **2014** (2014), 178. https://doi.org/10.1186/1687-6180-2014-178
8. M. Lopes, Unknown sparsity in compressed sensing: denoising and inference, *IEEE Trans. Inform. Theory*, **62** (2016), 5145–5166. https://doi.org/10.1109/TIT.2016.2587772
9. C. An, J. Ran, Hard thresholding hyperinterpolation over general regions, *J. Sci. Comput.*, **102** (2025), 37. https://doi.org/10.1007/s10915-024-02754-4
10. C. An, J. Ran, A. Sommariva, Hybrid hyperinterpolation over general regions, *Calcolo*, **62** (2025), 3. https://doi.org/10.1007/s10092-024-00625-w
11. Z. He, Q. Shu, Y. Wang, J. Wen, A ReLU-based hard-thresholding algorithm for non-negative sparse signal recovery, *Signal Process.*, **215** (2024), 109260. https://doi.org/10.1016/j.sigpro.2023.109260
12. J. Xue, Y. Zhao, T. Wu, J. Chan, Tensor convolution-like low-rank dictionary for high-dimensional image representation, *IEEE Trans. Circ. Syst. Vid.*, **34** (2024), 13257–13270. https://doi.org/10.1109/TCSVT.2024.3442295
13. J. Xue, Y. Zhao, S. Huang, W. Liao, J. Chan, S. Kong, Multilayer sparsity-based tensor decomposition for low-rank tensor completion, *IEEE Trans. Neur. Net. Lear.*, **33** (2022), 6916–6930. https://doi.org/10.1109/TNNLS.2021.3083931
14. J. Xue, Y. Zhao, Y. Bu, J. Chan, S. Kong, When Laplacian scale mixture meets three-layer transform: a parametric tensor sparsity for tensor completion, *IEEE Trans. Cybernetics*, **52** (2022), 13887–13901. https://doi.org/10.1109/TCYB.2021.3140148
15. J. Huang, F. Zhang, J. Wang, X. Liu, J. Jia, The perturbation analysis of nonconvex low-rank matrix robust recovery, *IEEE Trans. Neur. Net. Lear.*, **35** (2024), 15710–15723. https://doi.org/10.1109/TNNLS.2023.3289209

16. J. Huang, F. Zhang, X. Liu, J. Wang, J. Jia, R. Wang, Stable recovery of sparse signals with non-convex weighted r-norm minus 1-norm, *J. Comput. Math.*, **43** (2025), 43–62. https://doi.org/10.4208/jcm.2307-m2022-0225
17. J. Huang, X. Liu, F. Zhang, G. Luo, R. Tang, Performance analysis of unconstrained l_p minimization for sparse recovery, *Signal Process.*, **233** (2025), 109937. https://doi.org/10.1016/j.sigpro.2025.109937
18. J. Aujol, C. Dossal, A. Rondepierre, Convergence rates of the heavy-ball method under the Lojasiewicz property, *Math. Program.*, **198** (2023), 195–254. https://doi.org/10.1007/s10107-022-01770-2
19. B. Polyak, Some methods of speeding up the convergence of iteration methods, *USSR Comp. Math. Math. Phys.*, **4** (1964), 1–17. https://doi.org/10.1016/0041-5553(64)90137-5
20. Y. Zeng, D. Han, Y. Su, J. Xie, On adaptive stochastic heavy ball momentum for solving linear systems, *SIAM J. Matrix Anal. Appl.*, **45** (2024), 1259–1286. https://doi.org/10.1137/23M1575883
21. H. Attouch, J. Fadili, From the Ravine method to the Nesterov method and vice versa: a dynamical system perspective, *SIAM J. Optimiz.*, **32** (2022), 2074–2101. https://doi.org/10.1137/22M1474357
22. Y. Nesterov, A method for unconstrained convex minimization problem with the rate of convergence $O(1/k^2)$, *Dokl. Akad. Nauk. SSSR*, **269** (1983), 543–547.
23. X. Zeng, J. Lei, J. Chen, Dynamical primal-dual Nesterov accelerated method and its application to network optimization, *IEEE Trans. Automat. Contr.*, **68** (2023), 1760–1767. https://doi.org/10.1109/TAC.2022.3152720
24. E. Ghadimi, H. Feyzmahdavian, M. Johansson, Global convergence of the heavy-ball method for convex optimization, *Proceedings of European Control Conference (ECC)*, 2015, 310–315. https://doi.org/10.1109/ECC.2015.7330562
25. S. Saab, S. Phoha, M. Zhu, A. Ray, An adaptive polyak heavy-ball method, *Mach. Learn.*, **111** (2022), 3245–3277. https://doi.org/10.1007/s10994-022-06215-7
26. B. Shi, S. Du, M. Jordan, W. Su, Understanding the acceleration phenomenon via high-resolution differential equations, *Math. Program.*, **195** (2022), 79–148. https://doi.org/10.1007/s10107-021-01681-8
27. B. Goujaud, A. Taylor, A. Dieuleveut, Provable non-accelerations of the heavy-ball method, arXiv: 2307.11291. https://doi.org/10.48550/arXiv.2307.11291
28. J. Wei, L. Chen, Accelerated over-relaxation heavy-ball method: achieving global accelerated convergence with broad generalization, arXiv: 2406.09772. https://doi.org/10.48550/arXiv.2406.09772
29. Z. Sun, J. Zhou, Y. Zhao, N. Meng, Heavy-ball-based hard thresholding algorithms for sparse signal recovery, *J. Comput. Appl. Math.*, **430** (2023), 115264. https://doi.org/10.1016/j.cam.2023.115264
30. D. Needell, J. Tropp, CoSaMP: iterative signal recovery from incomplete and inaccurate samples, *Appl. Comput. Harmon. Anal.*, **26** (2009), 301–321. https://doi.org/10.1016/j.acha.2008.07.002
31. W. Dai, O. Milenkovic, Subspace pursuit for compressive sensing signal reconstruction, *IEEE Trans. Inform. Theory*, **55** (2009), 2230–2249. https://doi.org/10.1109/TIT.2009.2016006

32. Y. Zhao, Z. Luo, Dynamic orthogonal matching pursuit for sparse data reconstruction, *IEEE Open Journal of Signal Processing*, **4** (2023), 242–256. https://doi.org/10.1109/OJSP.2023.3247301

33. J. Bouchot, S. Foucart, P. Hitczenko, Hard thresholding pursuit algorithms: number of iterations, *Appl. Comput. Harmon. Anal.*, **41** (2016), 412–435. https://doi.org/10.1016/j.acha.2016.03.002

34. J. Blanchard, J. Tanner, Performance comparisons of greedy algorithms in compressed sensing, *Numer. Linear Algebr.*, **22** (2015), 254–282. https://doi.org/10.1002/nla.1948

35. J. Blanchard, J. Tanner, K. Wei, CGIHT: conjugate gradient iterative hard thresholding for compressed sensing and matrix completion, *Inf. Inference*, **4** (2015), 289–327. https://doi.org/10.1093/imaiai/iav011

AIMS Press