Hindawi
Mathematical Problems in Engineering
Volume 2021, Article ID 6615584, 10 pages
https://doi.org/10.1155/2021/6615584

Research Article

An Improved Sparsity Adaptive Matching Pursuit Algorithm and Its Application in Shock Wave Testing

Jiahui Zhang,[1] Xiao Wang,[2] Mingchi Ju,[1] Tailin Han,[1] and Yingzhi Wang[1]

[1] *School of Electronics and Information Engineering, Changchun University of Science and Technology, Changchun 130022, China*
[2] *School of Biomedical Engineering, Hainan University, Haikou 570228, China*

Correspondence should be addressed to Yingzhi Wang; wyz@cust.edu.cn

Received 28 November 2020; Accepted 5 July 2021; Published 26 July 2021

Academic Editor: Zhiyun Lin

In the compressed sensing (CS) reconstruction algorithms, the problems of overestimation and large redundancy of candidate atoms will affect the reconstruction accuracy and probability of the algorithm when using Sparsity Adaptive Matching Pursuit (SAMP) algorithm. In this paper, we propose an improved SAMP algorithm based on a double threshold, candidate set reduction, and adaptive backtracking methods. The algorithm uses the double threshold variable step-size method to improve the accuracy of sparsity judgment and reduces the undetermined atomic candidate set in the small step stage to enhance the stability. At the same time, the sparsity estimation accuracy can be improved by combining with the backtracking method. We use a Gaussian sparse signal and a measured shock wave signal of the 15psi range sensor to verify the algorithm performance. The experimental results show that, compared with other iterative greedy algorithms, the overall stability of the DBCSAMP algorithm is the strongest. Compared with the SAMP algorithm, the estimated sparsity of the DBCSAMP algorithm is more accurate, and the reconstruction accuracy and operational efficiency of the DBCSAMP algorithm are greatly improved.

1. Introduction

In recent years, Candes, Donoho, and Tao have proposed a new theory of signal acquisition and processing-compressed sensing (CS) [1, 2]. This theory states that as long as a signal is sparse or sparse in a specific transform domain, a projection matrix that is incoherent with the transform basis can be used to project the high-dimensional sparse signal onto a low-dimensional space. And the original signal can be then reconstructed with high probability from these few projections containing enough reconstruction information by solving an optimization problem [3]. The CS theory only uses a lower sampling rate to randomly sample and compress signals, breaking through the traditional sampling theorem. It has great advantages in processing massive complex signals and is widely applied to various engineering practices. Shock wave signals are transient signals, which have the characteristics of short duration in the time domain and obvious beginning and end nodes [4]. Moreover, relatively centralized information is another advantage, that is to say, the information density is low in the whole acquisition process, so we can consider the existing of a transform domain which can be used for sparse representation of shock wave signals. Therefore, in this paper, we apply a shock wave signal to detect the performance of reconstruction algorithms in CS theory.

The research contents of CS theory can be divided into signal sparse representation, sampling matrix establishment, and reconstruction algorithm design. We focus on reconstruction algorithms in this paper. At present, the commonly used reconstruction algorithms mainly include convex optimization algorithms, combinatorial optimization algorithms, and iterative greedy algorithms [5]. Although convex optimization algorithms have good reconstruction effects, they are difficult to process large scale data as a consequence of its high time complexity. Combinatorial optimization algorithms have particularly high operational efficiency, but they have strict requirements on sampling structures and poor practicability. Only iterative greedy algorithms have small calculation amount, great reconstruction effects, easy implementation, and the widest application range.

Sparsity Adaptive Matching Pursuit (SAMP) algorithm is one of the iterative greedy algorithms. This algorithm solves the sparsity in signal reconstruction [6], breaking through the defect of using sparsity as prior information in traditional iterative greedy algorithms. However, SAMP algorithm has a fixed step-size, which easily affects the algorithm performance in the process of SAMP algorithm approaching the true sparsity by superimposing the step-size. The sparsity accuracy is higher if the step-size is too small, but the running time will increase accordingly. The operational efficiency can be improved if the step-size is too large, but the overestimation that reduces the reconstruction accuracy is easy to occur. Moreover, the large redundancy of the atomic candidate set in the later stage of the algorithm operational will affect the reconstruction probability.

In the improvement of SAMP algorithm, the most common method is the variable step-size method, which can make the step-size change more flexible. This method has been used in [7, 8]. In [7], the selection column of the inner product of atoms is expanded to 2 times by the combination of Compressive Sampling Matching Pursuit (CoSaMP) algorithm and SAMP algorithm. Although the atomic matching rate is improved, the sparsity is limited reducing the practicability of the algorithm. The improved SAMP algorithm in [8] integrates the regularization thought, which makes the atoms clipped twice, but also increases the running time of the algorithm. The Dice coefficient matching based SAMP (DSAMP) algorithm in [9] uses Dice coefficient to improve the atomic correlation formula and enhance the accuracy of the selected atoms. The method makes faster convergence of the algorithm, but the overall efficiency of the algorithm is not obvious. In addition, [10] refers to the Adaptive Iterative Forward-Backward (AFB) greedy algorithm proposed by the SAMP algorithm, which can select and remove atomic indexes in the candidate set. In order to accurately estimate the sparsity, the algorithm in [11] adopts a method of preestimating the sparsity, which can effectively reduce the number of iterations of the algorithm, but its effect is related to the signal type and cannot be widely used.

In this paper, we propose an improved SAMP algorithm to enhance the stability and balance the accuracy of sparsity estimation and the efficiency.

2. CS Theory and Reconstruction Algorithms

2.1. CS Theory. The core idea of compressed sensing theory is to project the sparse signal and reduce the redundant information, realizing the high probability reconstruction of original signal with fewer data.

The necessary condition for the application of compressed sensing theory is that the signal is sparse. If a signal x is sparse, it can be projected by the following linear transformation:

$$\mathbf{y} = \mathbf{\Phi x}, \tag{1}$$

where $\Phi \in \mathbf{R}^{M\times N}$ denotes the random projection matrix that can also be called the sampling matrix. And $\mathbf{y}$ is the M-dimensional measurements with $M << N$ data points. In other words, the length of the measurement signal is far less than that of the measured signal obtained.

It is necessary to make a sparse representation of $\mathbf{x}$ if $\mathbf{x}$ is a nonsparse signal. Let $\mathbf{\Psi} \in \mathbf{R}^{M\times N}$ be the orthonormal basis, and $\boldsymbol{\theta} \in \mathbf{R}^{N\times 1}$ is the sparse signal with the sparsity K, then the sparse representation process of signal $\mathbf{x}$ is as follows:

$$\mathbf{x} = \mathbf{\Psi}\boldsymbol{\theta}. \tag{2}$$

Combining (1) and (2) to obtain the nonsparse signal projection process,

$$\mathbf{y} = \mathbf{A}\boldsymbol{\theta}, \tag{3}$$

where $\mathbf{A} = \mathbf{\Phi\Psi}$ is the sensing matrix.

When $\mathbf{y}$ is received, if $\mathbf{A}$ satisfies the Restricted Isometry Property (RIP), as long as the number of measurements satisfies $M = \mathrm{O}(K\log(N/K))$, we can accurately reconstruct the original sparse signal. The reconstruction model is expressed as

$$\widehat{\theta} = \arg\min \ \|\boldsymbol{\theta}\|_0 \, s.t.\mathbf{y} = \mathbf{A}\boldsymbol{\theta}, \tag{4}$$

where $\widehat{\theta}$ is the sparse signal obtained by reconstruction, and $\| \, \|_0$ represents the ℓ_0 – norm, which is used to measure the nonconvex optimization value of signal sparsity. ℓ_0 – norm has high computational complexity and a large amount of numerical calculation, so it is difficult to achieve accurate data reconstruction. Generally, the ℓ_0 – norm minimization problem is converted into a simpler ℓ_1 – norm minimization problem for an equivalent solution. $\widehat{\theta}$ is defined as

$$\widehat{\theta} = \arg\min \ \|\boldsymbol{\theta}\|_1 \, s.t.\mathbf{y} = \mathbf{A}\boldsymbol{\theta}. \tag{5}$$

Except (5), there are other methods to achieve signal reconstruction by relaxing the ℓ_0 – norm to the ℓ_p – norm $(p \leq 1)$ or based on the Bayesian framework.

2.2. Reconstruction Algorithms. Reconstruction algorithm is one of the key technologies of compressed sensing. It solves the problem of how to use compressed sampling data to recover the original signal. The higher the accuracy of the reconstruction algorithm, the closer the reconstructed signal to the original. The solution of (4) belongs to the NP-hard problem, which can be solved by exhaustively enumerating the combinations of nonzero values in C_L^K. Therefore, a series of methods for solving suboptimal solutions have been proposed, and the most widely used is iterative greedy algorithm.

Iterative greedy algorithms select one or several atoms (column vectors) from the sampling matrix in each iteration to gradually match $\mathbf{y}$ or the residual of $\mathbf{y}$ until achieving the iteration stop condition. The earliest greedy iterative algorithms are Matching Pursuit (MP) algorithm and Orthogonal Matching Pursuit (OMP) algorithm. The solution idea is to update the candidate set by estimating one by one in each iteration. But the algorithms are susceptible to noise and atomic correlation [12], resulting in the low reconstruction efficiency. On this basis, many improved algorithms have appeared: Regularized OMP (ROMP) algorithm selects atoms carrying larger amounts of reconstruction information through a regularization process; CoSaMP algorithm selects $2K$ atomic indexes to update the candidate set in each iteration and then discards redundant atomic indexes by pruning; Subspace Pursuit (SP) algorithm is similar to CoSaMP algorithm, but each iteration updates the candidate set with K atomic indexes. Although these algorithms improve the reconstruction accuracy, efficiency, and robustness to a certain range, they all require the sparsity K as a prior for exact recovery. However, the piece of information cannot be obtained in advance in practical CS [13]. Therefore, when the sparsity is unknown, it is easy to cause overestimation or underestimation, which will affect the accuracy of signal reconstruction and the operational efficiency of algorithms. SAMP algorithm can realize signal reconstruction when solving the signal sparsity and overcome the defect of the aforementioned algorithms.

3. Sparsity Adaptive Matching Pursuit Algorithm and Its Improved Algorithm

3.1. SAMP Algorithm. SAMP algorithm combines the forward tracking of OMP algorithm and the backward tracking of SP algorithm. During the iteration process, the atomic candidate set is expanded with fixed step-size, and at the same time, bad atomic indexes are removed and new atomic indexes are added until the signal residual energy is reduced to 0 or a certain threshold [14]. SAMP algorithm can estimate sparsity and complete signal reconstruction when sparsity is unknown, and the stability of the algorithm is better than that of other algorithms.

Suppose a signal compressed sensing process: $\mathbf{y} = \mathbf{A}\boldsymbol{\theta}$, where $\mathbf{y}$ is the M-dimensional measurement signal, $\mathbf{A}$ is the $M \times N$-dimensional sensing matrix, and $\boldsymbol{\theta}$ is the N-dimensional sparse signal of the original signal. The main steps of the algorithm are represented as Algorithm 1 in Table 1.

Table 1. The main steps of the SAMP algorithm are as follows.

The computational complexity of the SAMP algorithm is mainly reflected in the solution of the least-squares problem in the loop body, and the formula $(\mathbf{A}_{\mathbf{C}_t}^{\mathbf{T}} \mathbf{A}_{\mathbf{C}_t})^{-1} \mathbf{A}_{\mathbf{C}_t}^{\mathbf{T}}$ is analyzed step by step [15], where $\mathbf{A}_{\mathbf{C}_t}, \in \boldsymbol{R}^{M \times S}$.

(1) The computational complexity of $\mathbf{A}_{\mathbf{C}_t}^{\mathbf{T}} \mathbf{A}_{\mathbf{C}_t}$ is O (SM^2).

(2) The computational complexity of $(\mathbf{A}_{\mathbf{C}_t}^{\mathbf{T}} \mathbf{A}_{\mathbf{C}_t})^{-1}$ is O (S^3).

(3) The computational complexity of $(\mathbf{A}_{\mathbf{C}_t}^{\mathbf{T}} \mathbf{A}_{\mathbf{C}_t})^{-1} \mathbf{A}_{\mathbf{C}_t}^{\mathbf{T}}$ is O (MS^2).

(4) $(\mathbf{A}_{\mathbf{C}_t}^{\mathbf{T}} \mathbf{A}_{\mathbf{C}_t})^{-1} \mathbf{A}_{\mathbf{C}_t}^{\mathbf{T}}$ is executed at most M times in the loop body, and the computational complexity is O $(M^2 S^2)$.

(5) In summary, the computational complexity of the SAMP algorithm is O $(M^2 S^2)$.

3.2. Improved Algorithm. Although SAMP can achieve reconstruction under the condition of unknown sparsity, the step-size for determining the update of the atomic candidate set in the algorithm is fixed, and there is a situation of overestimation or low operational efficiency. Besides, there are too many redundant atomic indexes in the candidate set at the later stage of the algorithm, which is not conducive to algorithm reconstruction. To solve the above problems, we propose an improved algorithm. We adjust the step-size by setting double threshold parameters to improve the accuracy of sparsity estimation and filter the candidate set in the small step-size stage to improve the reconstruction probability. At the same time, we reduce the number of atomic indexes in the candidate set through the backtracking of the overall state of the algorithm, which improves the estimation accuracy of sparsity. In order to conveniently express the improved algorithm in this paper, combined with the improved methods, it is named DCBSAMP (D-Double Threshold; C-Candidate set reduction; B-Backtracking) algorithm. This abbreviation is used to represent the improved algorithm in the following paper.

3.2.1. Double Threshold Method. Reference [16] points out that the double threshold method proposed according to the variation of residual energy can adaptively solve the sparsity K, which can improve the reconstruction speed whereas ensuring high reconstruction accuracy. Its basic idea is to approach K quickly with large steps and gradually approach K with small steps [11]. In the algorithm, a double threshold judgment condition for residual energy is designed. One large threshold Th1 is set in the judgment condition 1. When the residual energy is close to Th1, it means that the algorithm quickly approaches K with large steps, which can reduce the reconstruction time. The other threshold Th2 (Th2 << Th1) is set in the judgment condition 2. When the residual energy is close to Th2, it means that the algorithm gradually approaches K through small steps to improve the reconstruction accuracy. The algorithm adaptively changes the step-size S according to the residual energy of the signal, thereby adjusting the value of the candidate set length L, reducing overestimation, and improving the efficiency of the algorithm. In this paper, we select parabolic function $y = \sqrt{x}$ and logarithmic function $y = \ln(x)$ as the step change model [17], and the specific step update formula is

$$\begin{cases} S = \text{round}\left(\sqrt{S_0^2 + \text{Stage}}\right), \|\mathbf{r_t}\|_2 > \text{Th1}, \\ S = \text{round}\left(a \ln\left(S_0 + \text{Stage}\right)\right), \text{Th2} \leq \|\mathbf{r_t}\|_2 \leq \text{Th1}, \end{cases} \tag{6}$$

where $S_0 = M/\sqrt{N}$, Stage is the number of step update times, round () means rounding, $\|\mathbf{r_t}\|_2$ is the residual energy, and a is a constant.

3.2.2. Candidate Set Reduction. The SAMP algorithm does not perform secondary screening of atomic indexes when forming the total candidate set $\mathbf{C}$, and a large number of redundant atomic indexes will enter each iteration [18]. If the number of atomic indexes in $\mathbf{C}$ is greater than the number of measurements M, the algorithm will end the iterative process before the signal fails to meet the conditions for successful reconstruction. And as L increases in the later stage, the number of redundant atomic indexes will also

```
Input: sensing matrix $\mathbf{A}$, the measurement signal $\mathbf{y}$, step-size $S$.
    Initialize: $t = 0$, $\mathbf{r_0} = \mathbf{y}$, $\mathbf{\Lambda_0} = \varnothing$, $L = S$.
    Repeat
(1) $\mathbf{J_t} = max\ \{\lambda_j = |\langle \mathbf{r_{t-1}}, \mathbf{a_j} \rangle|, L\}$. Seek the index of the first $L$ most matching atoms, where $\mathbf{J}$ is the pending candidate set of atoms.
(2) $\mathbf{C_t} = \mathbf{\Lambda}_{t\text{-}1} \cup \mathbf{J_t}$. Construct the total candidate set of atoms.
(3) $\overline{\theta}_\mathbf{t} = (\mathbf{A}_{\mathbf{C_t}}^{\mathbf{T}} \mathbf{A}_{\mathbf{C_t}})^{-1} \mathbf{A}_{\mathbf{C_t}}^{\mathbf{T}} \mathbf{y}$. Solve the least-squares problem. And according to $\mathbf{F_t} = max\ \{|\overline{\theta}_{\mathbf{t_j}}|, L\}$, find the index of the first $L$ best atoms from $\mathbf{C_t}$, where $\mathbf{F}$ is the atom construction set.
(4) $\mathbf{r_t} = \mathbf{y} - \mathbf{A}_{\mathbf{F_t}} \overline{\theta}_{\mathbf{F_t}}$. Calculate the residual.
(5) If $\|r_t\|_2 < 1e - 6$, the algorithm iteration stop condition is satisfied, output $\widehat{\theta} = \mathbf{A}_{\mathbf{F_t}}^{\mathbf{T}} \times \mathbf{y}$, $\widehat{\mathrm{K}} = \mathrm{L}$; otherwise, go to 6.
(6) If $\|r_t\|_2 \geq \|r_{t\text{-}1}\|_2$, $Stage = Stage + 1$, $L = Stage \times S$, go to 1; otherwise, $\mathbf{\Lambda_t} = \mathbf{F_t}$, $\mathbf{r_{t-1}} = r_t$, $t = t + 1$, go to 1.
Until iteration stop condition is true.
Output: estimated sparse signal $\widehat{\theta}$; estimated signal sparsity $\widehat{K}$.
```

Algorithm 1: The SAMP algorithm.

increase. According to the method of introducing fuzzy threshold in the preselection stage of a candidate set in [19], under the condition that the algorithm can be iterated normally, if the number of redundant atomic indexes in **C** in the later stage can be effectively reduced, the stability of the algorithm can be enhanced. What is more, since the effective atoms carrying more reconstruction information are more dominant in matching with the residual, the atomic indexes will appear in the upper part of the pending candidate set **J**. Therefore, in this paper, after the algorithm enters the small step stage, the size L of **J** is appropriately reduced, and we have

$$L1 = (\delta - \xi \times \text{flag2})L, \text{Th2} < \|\mathbf{r_t}\|_2 < \text{Th1}. \tag{7}$$

$L1$ is the size of the reduced **J**, the parameter $\delta \in (0.5, 0.9)$, the parameter $\xi << \delta$, and the flag bit flag2 is the number of times the algorithm enters in the small step stage. At this time, there will be no failure of algorithm reconstruction due to an insufficient number of basic atomic indexes in **C**. Furthermore, the expansion speed of **C** will slow down, and the algorithm will get more iterative opportunities to find the final set of atoms and improve the reconstruction probability.

3.2.3. Algorithm Backtracking. Although the double threshold variable step-size method can improve the estimation accuracy of sparsity, the algorithm still has shortcomings: it is impossible to effectively compare the estimated sparsity of the algorithm with the true sparsity when the signal sparsity K is unknown. If the algorithm is updated to the last step when S is too large, it is still prone to overestimation [20]. Based on this, we propose a method that can backtrack the overall operational state of the algorithm to determine whether the signal sparsity is overestimated and to further approximate the true sparsity. Save but not output the results when reaching the normal iteration stop condition for the first time. The overall algorithm is traced back to the previous iteration state according to the stored parameters. That is, $\mathbf{F_t} = \mathbf{F_{t-1}}$, $L = L - S$, $\mathbf{r_{t-1}} = \mathbf{r_{t-2}}$, S is reduced to $S1$, and after using the update method of $S1$, let $L = L + S1$. And then it iterates normally until reaching the iteration stop condition for the second time.

Compare the sparsity of the two outputs and take the smaller one and its corresponding reconstructed signal output. The estimated sparsity obtained is closest to the true sparsity. Use flag bit flag1 to determine the number of times that the algorithm normally reaches the iteration stop condition: when flag1 = 0, the algorithm reaches the iteration stop condition for the first time; when flag1 = 1, the algorithm reaches the iteration stop condition for the second time.

3.2.4. DCBSAMP Algorithm Steps. The main steps of the DCBSAMP algorithm are represented as Algorithm 2 in Table 2 and the algorithm flow chart is shown in Figure 1.

Table 2. The main steps of the DCBSAMP algorithm.

The adaptive change of the algorithm step length and the reduction of the redundancy of the candidate set can be realized through the above steps. As a result, the algorithm performance can be effectively improved. And the computational complexity of the DCBSAMP algorithm is mainly reflected in the solution of the least-squares problem in the loop body. When the worst-case execution times of the loop body is M, the computational complexity is $O\ (M^2 S^2)$, which is equal to the SAMP algorithm.

4. Experimental Results and Discussion

To verify the performance of the DCBSAMP algorithm, we use a Gaussian sparse signal and a shock wave signal to conduct experiments. And we select SAMP algorithm, OMP algorithm, DSAMP algorithm in [9], AFB algorithm in [10], and SP algorithm as control groups.

4.1. A Gaussian Sparse Signal to Algorithm Performance Index Test. Gaussian sparse signals are random compared with shock wave signals. They have controllable sparsity and the nonzero amplitude of the signals obeys the Gaussian distribution. They are commonly used experimental signals when testing the stability of reconstruction algorithms. Suppose the length of the all-zero signal **x** is N, and any K ($K << N$) element index forms the matrix **H**, then the Gaussian sparse signal is

```
Input: sensing matrix $\mathbf{A}$, the measurement signal $\mathbf{y}$, step-size $S$.
Initialization: $\mathbf{r_0} = \mathbf{y}$, $\mathbf{\Lambda_0} = \varnothing$, $L = S$, $t = 1$, $Stage = 0$, $flag1 = 0$, $flag2 = 0$, $S_0 = M/\sqrt{N}$.
Repeat
  (1) If $flag2 = 0$, $\boldsymbol{J_t} = max\,\{\lambda_j = |\langle \mathbf{r_{t-1}}, \mathbf{a_j} \rangle|, L\}$, search for the most matching first $L$ atomic indexes; if $flag2 > 0$, $\boldsymbol{J_t} = max\,\{\lambda_j = |\langle \mathbf{r_{t-1}}, \mathbf{a_j} \rangle|,$, $\mathrm{L1} = (\delta - \xi \times \mathrm{flag2})\mathrm{L}\}$, select the first $L1$ atomic indexes.
  (2) $\mathbf{C_t} = \mathbf{\Lambda_{t-2}} \cup \mathbf{J_t}$.
  (3) $\overline{\theta}_\mathbf{t} = (\mathbf{A}_{\mathbf{C_t}}^\mathbf{T}\ \mathbf{A}_{\mathbf{C_t}})^{-1} \mathbf{A}_{\mathbf{C_t}}^\mathbf{T}\ \mathbf{y}$, $\mathbf{F_t} = max\,\{|\overline{\theta}_\mathbf{tj}|, L\}$. Store the first $L$ $(L1)$ optimal atomic indexes in $\mathbf{C_t}$ into $\mathbf{F_t}$.
  (4) $\mathbf{r_t} = \mathbf{y} - \mathbf{A}_{\mathbf{F_t}} \overline{\theta}_{\mathbf{F_t}}$. Calculate the residual.
  (5) If $\|\mathbf{r_t}\|_2 \geq Th2$, go to 6; otherwise, judge whether $flag1 = 0$ is satisfied. If it is satisfied, set $flag1 = 1$, $\mathbf{\Lambda_t} = \mathbf{F_t} = \mathbf{F_{t-1}}$, $\mathbf{r_{t-1}} = \mathbf{r_{t-2}}$, $L = L - S1$, $t = t + 1$ and go to 1; if not, output $\widehat{\theta} = \mathbf{A}_{\mathbf{F_t}}^\mathbf{T} \times \mathbf{y}$, $\widehat{\mathrm{K}} = \mathrm{L}$.
  (6) If $\|\mathbf{r_t}\|_2 \leq Th1$, go to 7; otherwise, judge whether $\|\mathbf{r_t}\|_2 \geq \|\mathbf{r_{t-1}}\|_2$ is satisfied. If it is satisfied, there are $Stage = Stage + 1$, $S = round\,(\sqrt{S_0^2 + Stage}) = S1$, $t = t + 1$, and if $flag1 = 1$, make $S1 = 0.45S$, $L = L + S1$, go to 1; otherwise, there are $\mathbf{\Lambda_t} = \mathbf{F_t}$, $\mathbf{r_{t-1}} = \mathbf{r_t}$, $t = t + 1$, go to 1.
  (7) $_{-\mathbf{1}}flag2 = flag2 + 1$, if $\|\mathbf{r_t}\|_2 \geq \|\mathbf{r_{t-1}}\|_2$, $Stage = Stage + 1$, $S = round\,(a \ln (S_0 + Stage)) = S1$, $t = t + 1$, go to 1, and if $flag1 = 1$, so $S1 = 0.2S$, $L = L + S1$, go to 1; otherwise, $\boldsymbol{\Lambda_t} = \boldsymbol{F_t}$, $\boldsymbol{r_{t\text{-}1}} = \boldsymbol{r_{t\text{-}2}}$, $t = t + 1$, go to 1.
Until iteration stop condition is true.
Output: Estimated sparse signal $\widehat{\theta}$; estimated signal sparsity $\widehat{K}$.
```

Algorithm 2: The DCBSAMP algorithm.

$$\mathbf{x_H} = \rho \sum_{i=1}^{\mathrm{K}} \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{\mathbf{x}_{\mathbf{H_i}}^2}{2} \right). \tag{8}$$

Among them, ρ represents any constant.

The purpose of compressed sensing is to use the smallest possible number of measurements M to realize the reconstruction of the N-dimensional original signal. To compare the change results of the reconstruction probability and reconstruction error of each algorithm under the condition of the different number of measurements M, we design experiment 1.

Experiment 1: A Gaussian sparse signal is of length $N = 256$ and the sparsity is fixed as $K = 20$. The sampling matrix is the Bernoulli random matrix. The different numbers of measurements are chosen from $M = 50$ to $M = 115$ and for each M, 1000 simulations are conducted to calculate the probabilities of exact reconstruction for DCBSAMP, SAMP ($S = 16$) ($S = 8$), OMP, DSAMP, AFB, and SP algorithms. When the residual energy of algorithms is less than $1e-6$, the reconstruction is regarded as successful. Observe reconstruction probability and error.

As can be seen from Figure 2, the reconstruction probability of the DCBSAMP algorithm is always optimal from $M = 50$ to $M = 115$, and the stability is improved by 11.18% compared with the optimal small step SAMP algorithm. When $M < 80$, the reconstruction probability of each algorithm does not reach 100%, but the reconstruction probability of the DCBSAMP algorithm is the best. When $M = 80$, the DCBSAMP algorithm can already achieve 100% reconstruction probability. Among other algorithms, the SP algorithm achieves 100% reconstruction probability when $M = 85$, and the AFB algorithm achieves 100% reconstruction probability when $M = 95$. The OMP algorithm still cannot achieve 100% reconstruction probability when $M = 110$.

As shown in Figure 2, when $M \leq 70$, the reconstruction probability of all algorithms is less than 90%, which cannot be used in actual projects. Combined with Figure 3 focusing on the case of $M > 70$, the DCBSAMP algorithm is the algorithm with the smallest reconstruction error from $M = 55$ to $M = 115$. Especially after $M \geq 80$, the reconstruction probability of the algorithm reaches 100%. It can be seen that the DCBSAMP algorithm has the minimum reconstruction error whereas ensuring the reconstruction probability. The average error is about 10.85% lower than that of the small step SAMP algorithm, which effectively improves the reconstruction probability and accuracy of the compressed sensing system.

As shown in Figures 2 and 3, the DCBSAMP algorithm has higher reconstruction probability and lower reconstruction error, but compared with the large step SAMP algorithm and the DSAMP algorithm with similar experimental results, the performance advantage of the DCBSAMP algorithm is smaller. It can be approximately considered that the DCBSAMP algorithm based on the SAMP algorithm does not reduce the algorithm performance when the number of measurements M changes. In Experiment 1, we find that the results of DCBSAMP, SAMP ($S = 8$) and DSAMP algorithms are very closed; however, SAMP ($S = 8$) and DSAMP algorithms are both at the expense of more computing resources. Further it is proved by the algorithm runtime comparison; that is, the DCBSAMP algorithm consumes less calculation time to achieve better reconstruction results.

For original signals, the sparsity of different signal segments may be different. Using the largest possible sparsity K to realize the reconstruction of the N-dimensional original signal can detect the effectiveness of algorithms. To compare the change results of the reconstruction probability of each algorithm in the case of different sparsity K, we design experiment 2.

Experiment 2: a Gaussian sparse signal is of length $N = 256$ and the number of measurements is fixed as $M = 128$. The sampling matrix is a Bernoulli random matrix. Different sparsity is chosen from $K = 10$ to $K = 75$. This

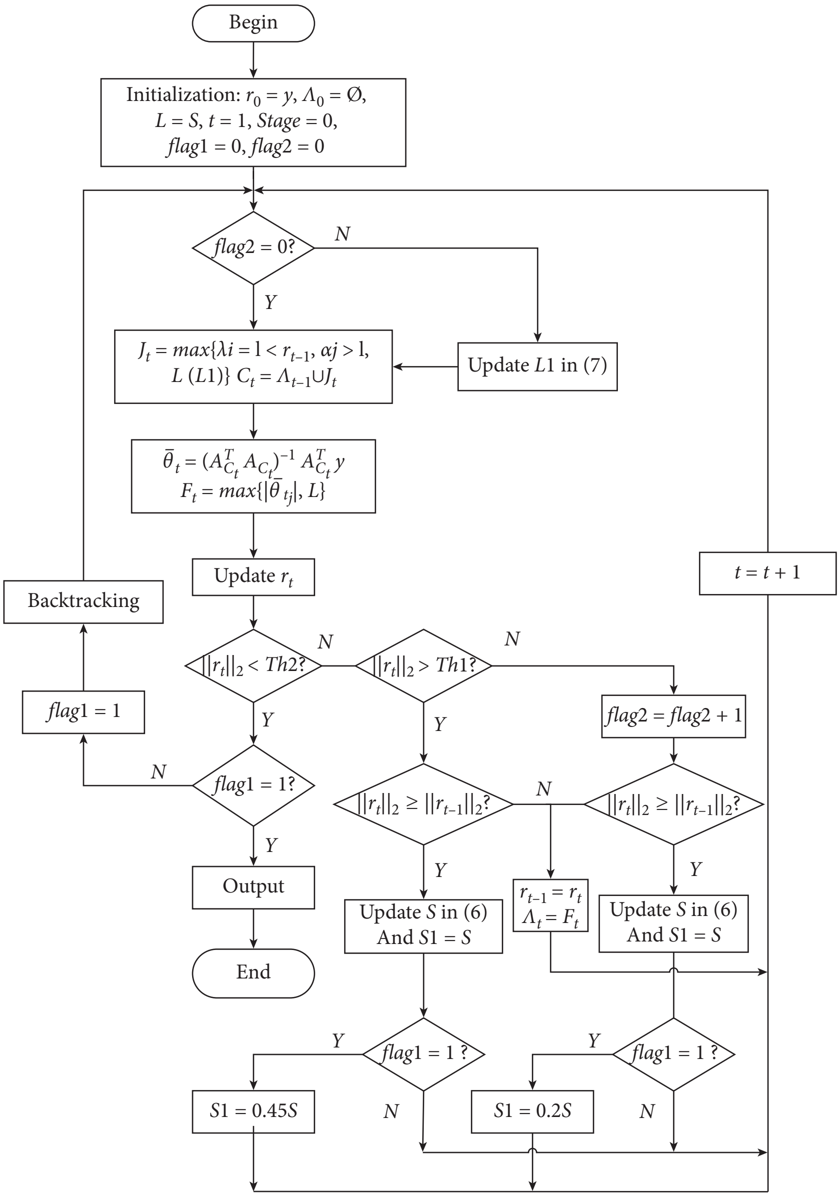

Begin
Initialization: $r_0 = y$, $\Lambda_0 = \varnothing$, $L = S$, $t = 1$, $Stage = 0$, $flag1 = 0$, $flag2 = 0$
$flag2 = 0$?
N
Y
$J_t = max\{\lambda i = 1 < r_{t-1}, \alpha j > 1, L\ (L1)\}$ $C_t = \Lambda_{t-1} \cup J_t$
Update $L1$ in (7)
$\bar{\theta}_t = (A_{C_t}^T A_{C_t})^{-1} A_{C_t}^T y$
$F_t = max\{|\bar{\theta}_{tj}|, L\}$
Update r_t
Backtracking
$t = t + 1$
$\|r_t\|_2 < Th2$?
N
$\|r_t\|_2 > Th1$?
N
Y
Y
$flag1 = 1$
$flag2 = flag2 + 1$
N
$flag1 = 1$?
$\|r_t\|_2 \geq \|r_{t-1}\|_2$?
N
$\|r_t\|_2 \geq \|r_{t-1}\|_2$?
Y
Y
Y
Output
Update S in (6) And $S1 = S$
$r_{t-1} = r_t$
$\Lambda_t = F_t$
Update S in (6) And $S1 = S$
End
Y
$flag1 = 1$?
Y
$flag1 = 1$?
$S1 = 0.45S$
N
$S1 = 0.2S$
N

Figure 1: The flow chart of the DCBSAMP Algorithm (flag1 is a flag bit for judging how many times the algorithm reaches the iteration stop condition: when flag1 = 0, the algorithm reaches the iteration stop condition for the first time; when flag1 = 1, the algorithm reaches the iteration stop condition for the second time).

procedure is repeated 1000 times for each value of K. Use OMP, SP, SAMP ($S = 16$) ($S = 8$), DSAMP, AFB, and DCBSAMP algorithms to reconstruct the signal and observe its reconstruction probability.

Figure 4 shows that the reconstruction probability of the DCBSAMP algorithm from $K = 10$ to $K = 75$ is always the best, and the stability is improved by 15.43% compared with the optimal algorithm. When $K \geq 65 > M/2$, the reconstruction probability attenuates significantly, and all algorithms cannot maintain the reconstruction probability above 90%. When $K < 65$, the reconstruction probability of the DCBSAMP algorithm is better than 90%. Only the SAMP algorithm with small steps and DSAMP algorithm can maintain the reconstruction effect close to the algorithm in this paper at the expense of algorithm efficiency, and the reconstruction probability of the DCBSAMP algorithm is always better than that of the two algorithms. The reconstruction probability of the other five algorithms attenuates earlier as the K value increases.

The experimental results shown in Figure 4 indicate that the DCBSAMP algorithm has obvious advantage in reconstruction probability compared with other algorithms when the sparsity K changes. In other words, when the signal sparsity is high, the DCBSAMP algorithm still has a large probability to reconstruct the signal, and the application range of the algorithm is wider. In order to further verify the engineering practicality of the algorithm, this paper uses a measured shock wave signals to further explore the reconstruction error and operational efficiency of the algorithm.

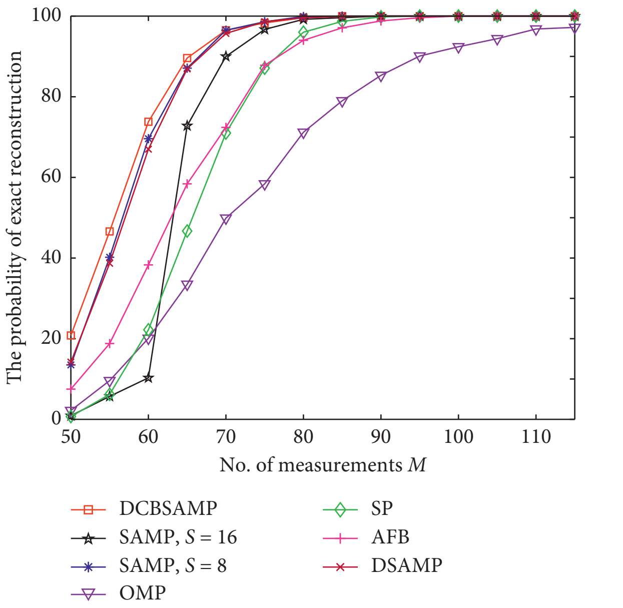


Figure 2: The Prob. of exact reconstruction vs. the number of measurements M. Here, the test signal is of length $N = 256$ and the sparsity is fixed as $K = 20$.

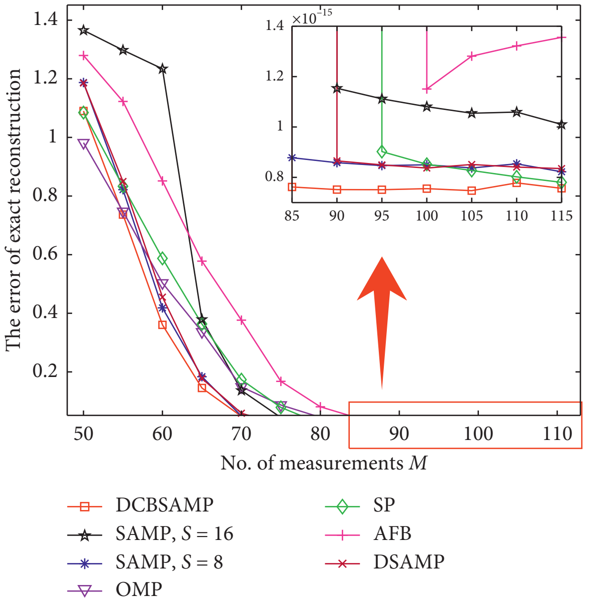


Figure 3: The error of exact reconstruction vs. the number of measurements M. Here, the test signal is of length $N = 256$ and the sparsity is fixed as $K = 20$.

4.2. A Shock Wave Signal to Algorithm Performance Index Test. Intercept a shock wave signal measured by the 15 psi range sensor and perform preprocessing such as precise interception and frequency reduction to obtain the shock wave signal with a length of $N = 4096$. Use the discrete wavelet matrix to sparse the signal to get the wavelet domain signal [4]. The time domain and wavelet domain of the shock waves are depicted in Figure 5.

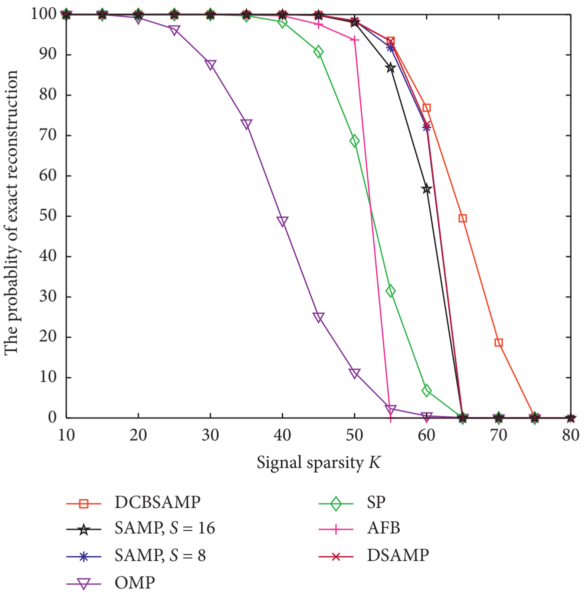
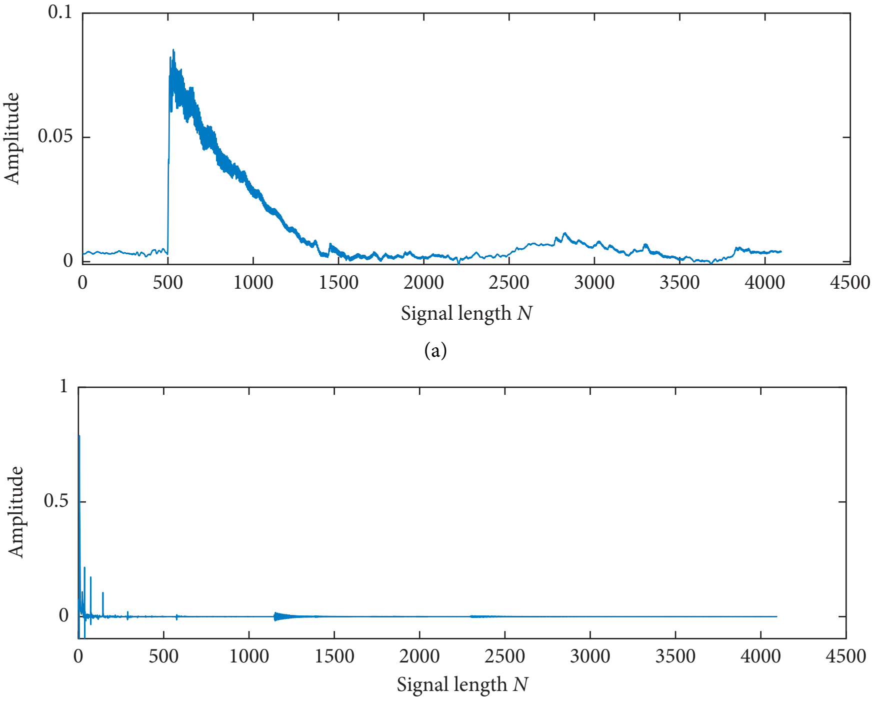


Figure 4: The Prob. of exact reconstruction vs. the signal sparsity K. Here, the test signal is of length $N = 256$ and the number of measurements is fixed as $M = 128$.

The DCBSAMP algorithm is an improved algorithm based on the SAMP algorithm, which belongs to sparsity adaptive algorithms. The operational efficiency and sparsity estimation accuracy of the DCBSAMP algorithm are affected by the true sparsity and step-size of the signal, which is different from other algorithms relying on sparsity. To compare the different results of the running time, the estimation accuracy of the sparsity, and the reconstruction error in the case of variable sparsity K of the adaptive sparsity algorithm, we design experiment 3.

Experiment 3: intercept a shock wave wavelet domain signal which is of length $N = 4096$ and the number of measurements is fixed as $M = 2048$. The sampling matrix is the Bernoulli random matrix. The sparsity K is generated by signal amplitude ft less than $0.0008 : 0.0003 : 0.0041$. This procedure is repeated 45 times for each value of K. Use DCBSAMP, DSAMP, AFB, and SAMP $(S = 16)$ $(S = 8)$ algorithms respectively to reconstruct the signal, and observe its reconstruction probability. When the SP algorithm is used to reconstruct signals with large sparsity, the operational efficiency is extremely low, and the reconstruction error is also large. The robustness of the SP algorithm is relatively weak. In the actual shock wave test, the length of the signal N and the sparsity K are very large, so the SP algorithm is not suitable for the application. Therefore, only add the OMP algorithm as a control group to compare the impact of algorithms that rely on sparsity on running time.

Figure 6 depicts that the average running time of the DCBSAMP algorithm from $K = 152$ to $K = 732$ is the shortest and the algorithm has the highest efficiency. When $K = 732$, the advantage of the DCBSAMP algorithm is the most obvious, which is 59.3% higher than that of the small step SAMP algorithm and 23.6% higher than that of the large one. The running time of the OMP algorithm depends on the



(a)

(b)

Figure 5: Graphics of a shock wave signal in time domain and wavelet domain. Here, the test signal is of length $N = 4096$ and the number of measurements is fixed as $M = 2048$. (a) Time domain and (b) wavelet domain.

sparsity, and the number of iterations is fixed at K times. The running time of the OMP algorithm in Figure 6 is lower than the small step SAMP algorithm after $K > 396$. If the step-size of the sparsity adaptive algorithm is selected appropriately, the running time can be saved. However, the appropriate S value cannot be selected directly owing to the unknown signal. The larger the step-size selected by the SAMP algorithm in Figure 6 is, the shorter the running time is. Conversely, the smaller the step-size is, the longer the running time is. However, the running time of the large step SAMP algorithm in Figure 6 is always higher than that of the DCBSAMP algorithm.

As shown in Figure 7, the sparsity estimation accuracy is the highest during the process of the DCBSAMP algorithm from $K = 152$ to $K = 732$. The sparsity obtained by the DCBSAMP algorithm is equal to or approximately equal to the true sparsity, and the maximum error is only 2. The error of AFB and DSAMP algorithms is much higher than that of the DCBSAMP algorithm. Regardless of whether the SAMP algorithm chooses large steps or more advantageous small steps to solve the sparsity, the error is much higher than that of the DCBSAMP algorithm. Moreover, the sparsity error curves of the other algorithms fluctuate significantly, and their stability is much lower than that of the DCBSAMP algorithm.

We normalize the reconstruction time of the algorithm to increase the attribute characteristics of the data. Combining it with the overestimation error respectively to obtain the error result under the effect of the algorithm reconstruction time and the overestimation is helpful for more comprehensive evaluation of the performance of the algorithm. The error result is shown in Figure 8.

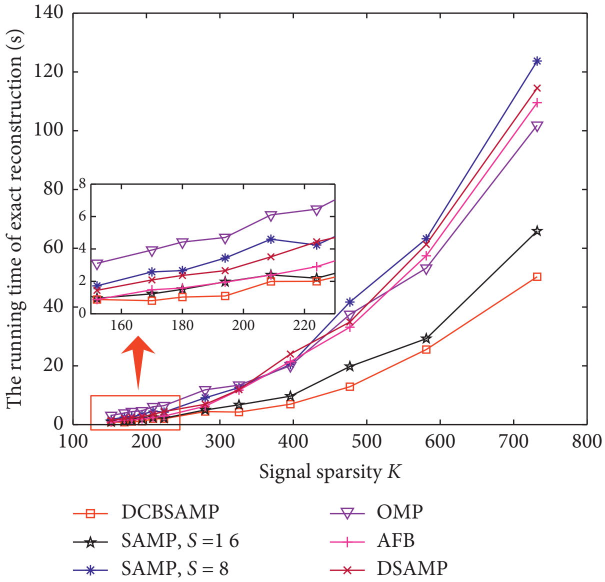


Figure 6: The running time of exact reconstruction vs. the signal sparsity K. Here, the test signal is of length $N = 4096$ and the number of measurements is fixed as $M = 2048$.

Figure 8 demonstrates that the average reconstruction error of the DCBSAMP algorithm from $K = 152$ to $K = 732$ is the smallest. When the operational efficiency gap between algorithms is large, the error is less affected by the

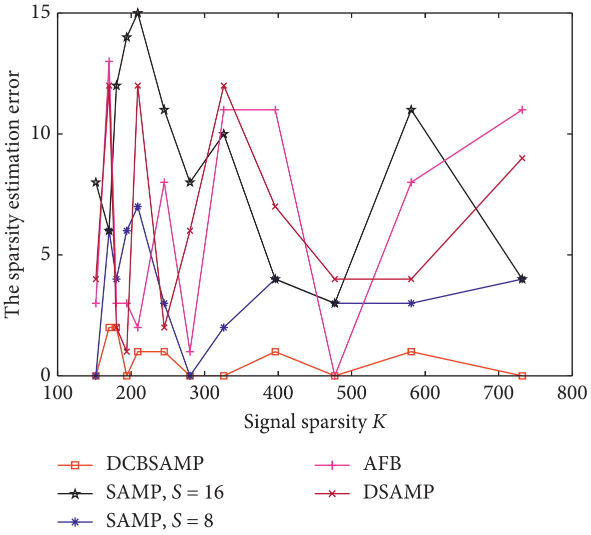


Figure 7: The sparsity estimation error vs. the signal sparsity K. Here, the test signal is of length $N = 4096$ and the number of measurements is fixed as $M = 2048$.

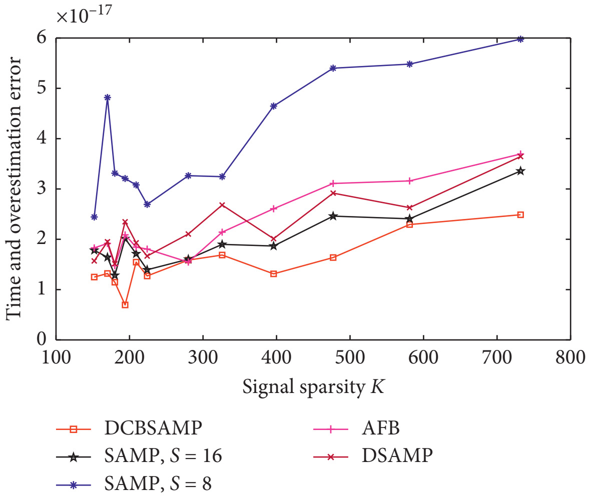


Figure 8: The time-overestimation error vs. the signal sparsity K. Here, the test signal is of length $N = 4096$ and the number of measurements is fixed as $M = 2048$.

sparsity estimation accuracy. On the contrary, when the efficiency gap is small, the error is greatly affected by the sparsity estimation accuracy. Although the sparsity estimation accuracy of the small step SAMP algorithm is lower than that of the large, the algorithm has the least operational efficiency, resulting in the biggest error. The DCBSAMP algorithm has the highest operational efficiency and sparsity estimation accuracy. The average error is reduced by 18.35% compared with the large step SAMP algorithm.

In summary, the DCBSAMP algorithm can seek a balance between operational efficiency and estimation effect, making the algorithm perform better than that of the SAMP, AFB, and DSAMP algorithms.

5. Conclusions

This paper proposes an improved SAMP algorithm, which combines the double threshold variable step-size method with the candidate set reduction method and the overall algorithm backtracking method to achieve an effective improvement in the stability and accuracy of the algorithm reconstruction. We use a Gaussian sparse signal and a shock wave signal of the 15 psi range sensor as test signals to verify the algorithm performance. The DCBSAMP algorithm is compared with traditional reconstruction algorithms to analyze the performance changes. Experiment results show the superior performance of the DCBSAMP algorithm over the existing algorithms. Compared with iterative greedy algorithms: when the number of measurements M changes, although the performance advantage of the algorithm is weak, the stability of the algorithm is still enhanced by 11.18%, and the reconstruction accuracy is also improved by 10.85% within the practical range of the algorithm more than that of the original optimal algorithm; when the sparsity K changes, the stability is enhanced by 15.43%. Compared with the SAMP algorithm and its improved algorithms: the operational efficiency of the algorithm has a maximum improvement of 59.3%. The maximum estimated sparsity error is only 2. And the time-overestimation error is also reduced by 18.35% when compared with the original optimal average.

Data Availability

The shock wave data used to support the findings of this study cannot be shared because they come from a confidential project.

Conflicts of Interest

The authors in this research declare that they have no conflicts of interest.

Authors' Contributions

X. W. proposed the framework of this research, carried out the main experiments, and modified the paper; J. Z. and M. J. carried out the rest of experiments and wrote the original paper; T. H. and Y. W. offered useful suggestions and supervised the research process.

Acknowledgments

This research was supported by the Project of the Science and Technology Department of Jilin Province in China under Grant 20200401116GX and Grant 20200602005ZP.

References

[1] D. L. Donoho, "Compressed sensing," *IEEE Transactions on Information Theory*, vol. 52, no. 4, pp. 1289–1306, 2006.

[2] E. J. Candès, "The restricted isometry property and its implications for compressed sensing," *Comptes Rendus Mathematique*, vol. 346, no. 9-10, pp. 589–592, 2008.

[3] D. L. Donoho, A. Maleki, and A. Montanari, “Message-passing algorithms for compressed sensing,” *Proceedings of the National Academy of Sciences*, vol. 106, no. 45, pp. 18914–18919, 2009.

[4] X. Wang, *Research of main techniques in distributed transient parameters wireless transmission system*, Ph.D. dissertation, CUST, Jilin, China, 2017.

[5] K. Wang, T. Han, Y. Gao, and X. Wang, “Shock wave signal acquisition based on improved Subspace tracking algorithm,” *Journal of Changchun University of Science & Technology*, vol. 42, pp. 89–94, 2019.

[6] D. L. Donoho, L. Can, N. Nguyen, and T. D. Tran, “Sparsity adaptive matching pursuit algorithm for practical compressed sensing in Pacific Grove,” *IEEE Asilomar*, pp. 581–587, 2008.

[7] L. Lei, G. Li, C. Jiang, and Z. Liang, “Sparsity variable step-size adaptive compressive sampling matching pursuit algorithm based On CS,” *Applied Soft Computing*, vol. 37, pp. 260–264, 2020.

[8] H. Liu, H. Zhong, and W. Feng, “Regularized sparsity variable step-size adaptive matching pursuit algorithm based on CS,” *J. B. University Aeronaut Astronaut*, vol. 43, pp. 2109–2117, 2017.

[9] M. D. Zhu, M. Li, Z. Geng et al., “Dice coefficient matching-based sparsity adaptive matching pursuit algorithm for the digital predistortion model pruning,” in *Proceedings of the 2018 18th IEEE International Conference on Communication Technology (ICCT 2018)*, pp. 1032–1035, Chongqing, China, October 2018.

[10] A. Aziz, W. Osamy, and A. M. Khedr, “Iterative selection and correction based adaptive greedy algorithm for compressive sensing reconstruction,” *Journal of King Saud University Computer and Information Sciences*, vol. 116, pp. 3277–3289, 2020.

[11] Y. Hu, “Research on signal reconstruction algorithms based on compressed sensing,” M. S. thesis, NEPU, Jilin, China, 2018.

[12] J. A. Tropp and A. C. Gilbert, “Signal recovery from random measurements via orthogonal matching pursuit,” *IEEE Transactions on Information Theory*, vol. 53, no. 12, pp. 4655–4666, 2007.

[13] Y. Chen, Y. Zhang, H. Liu et al., “FBG signal processing with improved multiple atom matching pursuit algorithm,” *Journal of Optoelectronics·Laser*, vol. 27, pp. 580–586, 2016.

[14] D. Zhang and H. Li, “Application of a variable step size SAMP algorithm in radar target recognition,” *Control Engineering*, vol. 25, no. 8, pp. 1381–1385, 2018.

[15] K. Yang, “Research on channel estimation based on compressed sensing,” M. S. thesis, CDUESTC, Sichuan, China, 2018.

[16] X. Du, X. Hu, B. Gu, B. Chen, and S. Qiu, “SAMP-RB compressed sensing reconstruction algorithm based on variable step,” *Journal of Applied Computing Research*, vol. 35, pp. 1084–1087, 2018.

[17] X. Bi, Z. Shang, Z. Qiang, and H. Liu, “Improvement of sparsity adaptive matching pursuit based on variable iteration steps,” *International Journal of Simulation Systems*, vol. 26, pp. 2116–2120, 2014.

[18] S. Zhang, M. Zhou, and C. Zhang, “Adaptive stepsize forward-backward matching pursuit algorithm,” *Applied Soft Computing*, vol. 33, pp. 244–248, 2016.

[19] T. Blumensath and M. E. Davies, “Stagewise weak gradient pursuits,” *IEEE Transactions on Signal Processing*, vol. 57, no. 11, pp. 4333–4346, 2009.

[20] R. Yang and X. Zhang, “An improved algorithm based on sparity adaptive matching pursuit,” *Journal of Nanjing University of Technology (Natural Science Edition)*, vol. 54, pp. 538–542, 2018.