

Block Sparse Signals Recovery Algorithm for Distributed Compressed Sensing Reconstruction

Xingyi Chen*, Yujie Zhang**, and Rui Qi***,****

Abstract

Distributed compressed sensing (DCS) states that we can recover the sparse signals from very few linear measurements. Various studies about DCS have been carried out recently. In many practical applications, there is no prior information except for standard sparsity on signals. The typical example is the sparse signals have block-sparse structures whose non-zero coefficients occurring in clusters, while the cluster pattern is usually unavailable as the prior information. To discuss this issue, a new algorithm, called backtracking-based adaptive orthogonal matching pursuit for block distributed compressed sensing (DCSBBAOMP), is proposed. In contrast to existing block methods which consider the single-channel signal reconstruction, the DCSBBAOMP resorts to the multi-channel signals reconstruction. Moreover, this algorithm is an iterative approach, which consists of forward selection and backward removal stages in each iteration. An advantage of this method is that perfect reconstruction performance can be achieved without prior information on the block-sparsity structure. Numerical experiments are provided to illustrate the desirable performance of the proposed method.

Keywords

Block Sparse Signals, Compressed Sensing, Distributed Compressed Sensing, Iteration Algorithm

1. Introduction

Distributed compressed sensing (DCS) is a signals recovery framework for acquiring sparse signals where both sensing and compression are performed at the same time. It uses sparsity of signals to recovery the signals from the measurements [1,2].

The DCS model can be stated as:

$$\mathbf{Y} = \Phi \mathbf{X}, \quad (1)$$

where $\Phi \in \mathbb{R}^{M \times N}$ is a random measurement matrix. This system model describes an under-determined system, where $M < N$. $\mathbf{Y} = [\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_J]$ are the measure vectors and $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_J]$ are unknown sparse vectors. The signal vector \mathbf{x}_i has $K^{(i)}$ non-zero components and indices with cardinality $|\text{supp}(\mathbf{x}_i)| = \|\mathbf{x}_i\|_0 = K^{(i)}$. The goal of DCS model is to reconstruct \mathbf{X} from \mathbf{Y} . The method

※ This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License (<http://creativecommons.org/licenses/by-nc/3.0/>) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.

Manuscript received August 11, 2017; first revision March 15, 2017; second revision May 17, 2018; accepted July 14, 2018.

Corresponding Author: Rui Qi (qqr0425@163.com)

* Faculty of Information Engineering, China University of Geosciences, Wuhan, China (rubycoc@163.com)

** School of Mathematics and Physics, China University of Geosciences, Wuhan, China (zhangyujie@cug.edu.cn)

*** School of Science, Naval University of Engineering, Wuhan, China (qqr0425@163.com)

****Hubei Subsurface Multi-scale Imaging Key Laboratory, China University of Geosciences, Wuhan, China (qqr0425@163.com)

is that

$$\mathbf{y}_i = \mathbf{A}\mathbf{x}_i \quad \text{subject to} \quad \min \|\mathbf{x}_i\|_0 \quad \forall i, \quad (2)$$

where $\|\mathbf{x}_i\|_0$ is the l_0 -norm of \mathbf{x}_i .

In practice, the DCS problem can be relaxed as an approximate convex optimization problem [3]:

$$\arg \min_{\{\mathbf{x}_i\}} \sum_{i=1}^K \|\mathbf{y}_i - \mathbf{A}\mathbf{x}_i\|_2^2 \quad \text{subject to} \quad \|\mathbf{x}_i\|_0 \leq K \quad \forall i. \quad (3)$$

Here, the parameter K states the maximum sparsity of the \mathbf{x}_i .

If the signals \mathbf{X} are independent, the reconstruction problem can be divided into J individual problems, we can reconstruct each signal independently by using the compressed sensing (CS) framework [4,5]. The more interesting case is when the signals $\mathbf{x}_i, i = 1, 2, \dots, J$ are correlated among each other. Then we can reconstruct \mathbf{X} jointly using DCS algorithms [6-10]. Moreover, it was shown that signals reconstruction based on DCS could save about 30% of measurements than using CS on each source [6].

Problem (3) is thought to be NP-hard [6]. Many pursuit algorithms [7-10] have been introduced to recover the signals with tractable complexity. It has been shown in paper [11] that l_1 -norm constraint is sufficient to ensure the sparsest solution for many high-dimensional cases.

DCS was defined by Duarte et al. [12] firstly. Two different joint sparse models (JSMs) were proposed by them. Subsequently, many algorithms were carried out. A new DCS algorithm was introduced in [7] and it exploited signal to signal correlation structures. Wakin et al. [13] proposed a simultaneous orthogonal matching pursuit (OMP) [14] method for DCS (named as jointOMP), which can reduce the number of measurements. Unfortunately, there is no backtracking mechanism in jointOMP, which leads to the worse recovery performance. To overcome the drawback, the subspace pursuit method for DCS (DCSSP) was proposed in [15]. A new joint sparse recovery method, called orthogonal subspace matching pursuit (OSMP), is proposed in [16]. Nevertheless, those algorithms have a common limitation that the signals sparsity must be pre-known, whereas it is usually unpractical in practice. Recently, two new recovery methods, named forward-backward pursuit method for DCS (DCSFBP) and backtracking-based adaptive OMP method for DCS (DCSBAOMP), are proposed [9,10]. In [17], l_1 / l_2 -norm is used to enforce joint sparsity on the signals. However, the above methods do not take into account the structures of signals or their representations. Sometimes, each signal \mathbf{x}_j under consideration is structured in nature, e.g., the structure of block sparse that the non-zero coefficients of signals occurring in cluster. The block-sparse signals appear in many applications including gene expression analysis [18] and equalization of sparse communication channels [19].

Block-sparse signal reconstruction algorithms were introduced and investigated in recent literatures. Here, we focus on the DCS of block-sparse signals. For a given matrix $\Phi \in \mathbb{R}^{M \times N}$, we reconstruct block-sparse signals \mathbf{X} from $\mathbf{Y} = \Phi\mathbf{X}$. Here \mathbf{X} with block size d can be formed as

$$\mathbf{X} = \begin{pmatrix} \mathbf{X}[1] \\ \mathbf{X}[2] \\ \vdots \\ \mathbf{X}[I] \end{pmatrix}, \quad (4)$$

where $\mathbf{X}[i] = (\mathbf{x}_{[(i-1)d+1]:id,1}, \mathbf{x}_{[(i-1)d+1]:id,2}, \dots, \mathbf{x}_{[(i-1)d+1]:id,J}) \in \mathbb{R}^{d \times J}$ denotes the i^{th} block matrix with length d and $N = ld$. The matrix \mathbf{X} is said to be K -block sparse if $\mathbf{X}[i]$ has nonzero Euclidean norm for at most K entries i . Denote

$$\|\mathbf{X}\|_{2,0} = \sum_{i=1}^l I(\|\mathbf{X}[i]\|_2 > 0),$$

where $I(\|\mathbf{X}[i]\|_2 > 0)$ is an indicator function. In this case, a block K -sparse matrix \mathbf{X} can be defined as $\|\mathbf{X}\|_{2,0} \leq K$ [20]. Then the total sparsity of \mathbf{X} is $K_{total} = K \times d \times J$. Similarly to (4), the measure matrix Φ is denoted as the following block form,

$$\Phi = \left(\underbrace{\Phi_1, \dots, \Phi_d}_{\Phi[1]}, \underbrace{\Phi_{d+1}, \dots, \Phi_{2d}}_{\Phi[2]}, \dots, \underbrace{\Phi_{N-d+1}, \dots, \Phi_N}_{\Phi[l]} \right), \tag{5}$$

here $\Phi[i]$ is a submatrix of size $M \times d$. The supports of the coefficient matrix \mathbf{X} are $\Gamma(\mathbf{X}) = \{i, \|\mathbf{X}[i]\|_2 \neq 0\}$ [20].

Here, the reconstruction problem can be relaxed as a convex optimization by

$$\arg \min_{\{\mathbf{X}[i]\}} \sum_{i=1}^K \left\| \mathbf{Y} - [\Phi[1], \dots, \Phi[l]] \begin{bmatrix} \mathbf{X}[1] \\ \vdots \\ \mathbf{X}[l] \end{bmatrix} \right\|_2^2 \quad \text{subject to } \|\mathbf{X}\|_{2,0} = \sum_{i=1}^l I(\|\mathbf{X}[i]\|_2 > 0) \leq K. \tag{6}$$

This problem is also a NP-hard problem. One natural idea is to replace the l_2 / l_0 -norm by l_2 / l_1 -norm, that is

$$\min \|\mathbf{X}\|_{2,1} \quad \text{subject to } \mathbf{Y} = \Phi \mathbf{X}, \tag{7}$$

where $\|\mathbf{X}\|_{2,1} = \sum_{i=1}^l \|\mathbf{X}[i]\|_2$.

Specifically, in [21], the block compressive sampling matching pursuit (BCoSaMP), which is based on the compressive sampling matching pursuit (CoSaMP) [22], was proposed. Eldar et al. [23] proposed the block version of the OMP (BOMP) algorithm for block-sparse signal recovery. In [24], a dictionary optimization problem for block-sparse signal reconstruction was investigated, but this method needed the maximum block length as the prior information. Another approaches, such as iterative hard thresholding (IHT) [21], block subspace pursuit (BSP) [25] and block StOMP (BStOMP) [26] have been investigated to date. However, most of the block algorithms deal with the single signal reconstruction.

Since DCS algorithms do not take into account the block structures of signals and most of the existing block compressed sensing algorithms just deal with the single-channel signal. In this paper, by incorporating the technique of DCS and the additional structure of block sparse signals, we propose a new recovery mechanism to solve the block DCS problem, called the backtracking-based adaptive OMP for block DCS (DCSBBAOMP) algorithm. In contrast to the most existing approaches, the new method can recover the block sparse signals simultaneously without the sparsity structure information.

Meanwhile, the new method can recover multiple sparse signals from their compressive measurements. Numerical experiments including recovery of random artificial sparse signals and the real-life data are provided to illustrate the desirable performance of the proposed method.

The rest of the paper is organized as follows: in the next section, the block sparse signals reconstruction method is presented. Experimental results are presented in Section 3 to compare the proposed algorithm with DCSBAOMP, backtracking-based adaptive OMP for block sparse signal (BBAOMP), DCSFBP and DCSSP. In Section 4, conclusions and future work are presented.

2. Proposed Algorithm for Block-Sparse Signals

As the extension of OMP, BBAOMP and DCSBAOMP were proposed by Qi et al. [27] and Zhang et al. [9], respectively. Those algorithms first find one or several atoms which corresponding to the much larger correlation between measurement vectors and residual. To be mentioned, the atoms chosen in the previous stage may be wrong. And then backtracking procedure is used to refine the estimated support set. In the next, these two methods obtain a new residual by using the least-square fit. BAOMP and DCSBAOMP methods do not need signals sparsity as a prior, they are repeated until the residual is smaller than a threshold or the iteration count reaches the number of maximum iteration. The main difference between BBAOMP and DCSBAOMP is that the BBAOMP algorithm is a single-channel method, which deals with block sparse signal, while DCSBAOMP algorithm is a multi-channel method. In this paper, we generalize the DCSBAOMP method to deal with the block sparse signals.

Generalizing the DCSBAOMP algorithm to the block sparse signals, we obtain the DCSBBAOMP algorithm. In this method, the measurement matrix Φ is considered to be a dictionary with each block column matrix $\Phi[i]$. It first chooses several block indices C^n whose correlations between the residual \mathbf{res}^{n-1} and the block columns $\Phi[i]$ are not smaller than $\mu_1 \cdot \max_{j \in \Omega_1} \|\langle \mathbf{res}^{n-1}, \Phi[j] \rangle\|_2$, $\Omega_1 = \{1, 2, \dots, l\}$, and then subtracts some wrong block indices Γ^n which corresponding the indices of $\|\mathbf{X}_F^n[i]\|_2$ are not larger than $\mu_2 \cdot \max_{j \in C^n} \|\mathbf{X}_F^n[j]\|_2$, the final estimated support set will be identified after several iterations. $\|\mathbf{X}\|_2$ returns the 2-norm of matrix \mathbf{X} , and it is equal to the largest singular value of \mathbf{X} . The details about the DCSBBAOMP algorithm are shown in Algorithm 1.

In this algorithm, μ_1 is a constant which determines the number of block indices chosen at each time. When $\mu_1=1$, only one block index is selected. When μ_1 becomes smaller, the DCSBBAOMP algorithm can select more than one block index at every iteration. Smaller μ_1 results in much more block indices are selected at each time and then speeds up the algorithm. Unfortunately, the block indices selected at the above process may be wrong. μ_2 is a parameter determining the number of deleted block indices. Similarly, bigger μ_2 results in smaller deleted block indices and slows down the algorithm. In many experiments, we find that the choice of μ_1 and μ_2 is same as those of BAOMP and DCSBAOMP. So in this paper, we do not discuss the change of μ_1 and μ_2 and choose the same values with BBAOMP and DCSBAOMP while $\mu_1=0.4$ and $\mu_2=0.6$. After updating the support set F , we generate a new residual by using the least-square fit. Due to the block sparsity K is not known in advance, the DCSBBAOMP algorithm is repeated until the residual \mathbf{res}^n is smaller than a threshold ε or the iteration count n reaches the number of maximum iterations n_{\max} .

Algorithm 1. DCSBBAOMP algorithmInput: $\Phi, Y, \mu_1, \mu_2, \varepsilon, n_{\max}$ Initialization: estimated support set $F = \emptyset$, initial solution $\mathbf{X}^0 = \mathbf{0}$,initial residual $\mathbf{res}^0 = \mathbf{Y}$, iteration count: $n = 1$ candidate set: $C^0 = \emptyset$, delete set: $\Gamma^0 = \emptyset$ while $\|\mathbf{res}^n\|_2 \geq \varepsilon$ and $n < n_{\max}$ doStep 1 (Forward): find the candidate set C^n that satisfying

$$\left\| \langle \mathbf{res}^{n-1}, \Phi[C^n] \rangle \right\|_2 \geq \mu_1 \cdot \max_{j \in \Omega_1} \left\| \langle \mathbf{res}^{n-1}, \Phi[j] \rangle \right\|_2, \Omega_1 = \{1, 2, \dots, l\}$$

Step 2: compute $F = F \cup C^n$, $\mathbf{X}_F^n = \Phi_F^\dagger \mathbf{Y}$ Step 3 (Backward): select the atom-delete set Γ^n in which the block indices i satisfying

$$\left\| \mathbf{X}_F^n[i] \right\|_2 \leq \mu_2 \cdot \max_{j \in C^n} \left\| \mathbf{X}_F^n[j] \right\|_2, i \in F$$

Step 4 (Update): $F = F \setminus \Gamma^n$, $n = n + 1$, $\mathbf{res}^n = \mathbf{Y} - \Phi_F \mathbf{X}_F^n$ Step 5 (Output): The estimation of sparse signals is $\tilde{\mathbf{X}}[\{1, 2, \dots, l\} - F] = 0$ and $\tilde{\mathbf{X}}[F] = \Phi_F^\dagger \mathbf{Y}$

3. Simulations and Results

In this section, several experiments are simulated to illustrate the performance of the proposed DCSBBAOMP method, which compared with DCSBAOMP, BBAOMP, DCSFBP and DCSSP algorithms.

In each trial, the block sparse signals \mathbf{X} are artificially generated as follows: for a fixed sparsity K , we randomly choose the nonzero blocks. Each element in the nonzero blocks is drawn from standard Gaussian distribution $N(0,1)$ and the elements of other blocks are zeros. The observation matrix is $\mathbf{Y} = \Phi \mathbf{X}$, where the entries of sensing matrix $\Phi \in \mathbb{R}^{M \times N}$ are generated from standard Gaussian distribution $N(0,1)$ independently. Firstly, we give two experiments including the changes of sparsity and number of measurements to compare the SNR and the run time of DCSBBAOMP with those of DCSBAOMP, BBAOMP, DCSFBP and DCSSP. Then we discuss the changes of reconstruction performance without the block size. Finally, the proposed method is tested on electrocardiography (ECG) signals.

To evaluate its performance, we use a measure named signal-to-noise ratio (SNR) defined as [10]:

$$\text{SNR} = 10 \log \left(\frac{\|\mathbf{X}\|_2^2}{\|\mathbf{X} - \hat{\mathbf{X}}\|_2^2} \right), \quad (8)$$

where \mathbf{X} denote the original signals and $\hat{\mathbf{X}}$ denote the reconstructed signals, respectively.

To be mentioned, each test is repeated 100 times. Then we compute the average values of SNR and running time, respectively. In the following experiments, the proposed method uses $\mu_1 = 0.4$, $\mu_2 = 0.6$, $n_{\max} = M$, $\varepsilon = 10^{-6}$ as the input parameters. For simplicity, in all the experiments, the sparse_value on x-axis is equal to K_{total} / J , that is,

$$\text{sparse_value} = \frac{K_{total}}{J}. \quad (9)$$

3.1 The Recovery Performance versus Sparsity with Small Sample

In the first experiment, the recovery performance is observed as the `sparse_value` varies from 10 to 70 with the fixed block size $d=2$ and $M=128$, $N=256$, $J=3$. The average SNR and run time are shown in the Fig. 1.

As can be seen from Fig. 1(a), the SNR of all the algorithms decreases slightly as the `sparse_value` varies from 10 to 70. DCSBAOMP and DCSBBAOMP algorithms obtain the similarly better SNR in here. When `sparse_value` > 60, the performance decreases significantly. The reason is that `sparse_value` is close to $M/2$, the recovery performance will fall down [28]. Fig. 1(b) depicts that the run time of all the algorithms increases as the `sparse_value` varies from 10 to 70. Specially, the DCSBAOMP gives much litter run time among the five algorithms. The reason lies in two aspects, one is that the DCSBAOMP algorithm is a multi-channel algorithm, which accelerates the speed of the algorithm; the other is that the DCSBAOMP utilizes the backtracking procedure, which leads to better reconstruction performance and shorter run time.

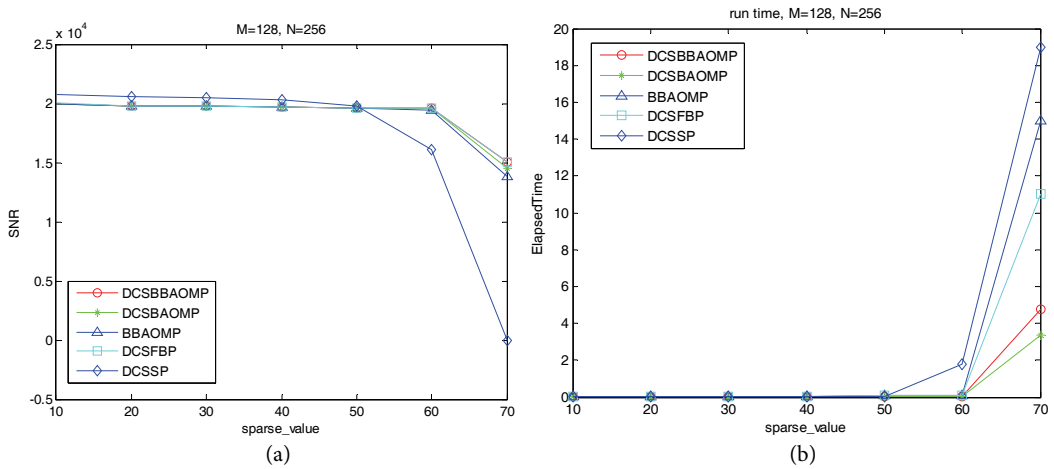


Fig. 1. Reconstruction results over the `sparse_value`. The numerical values on x-axis denote the `sparse_value` of signals and those on y-axis represent the SNR (a) and run time (b).

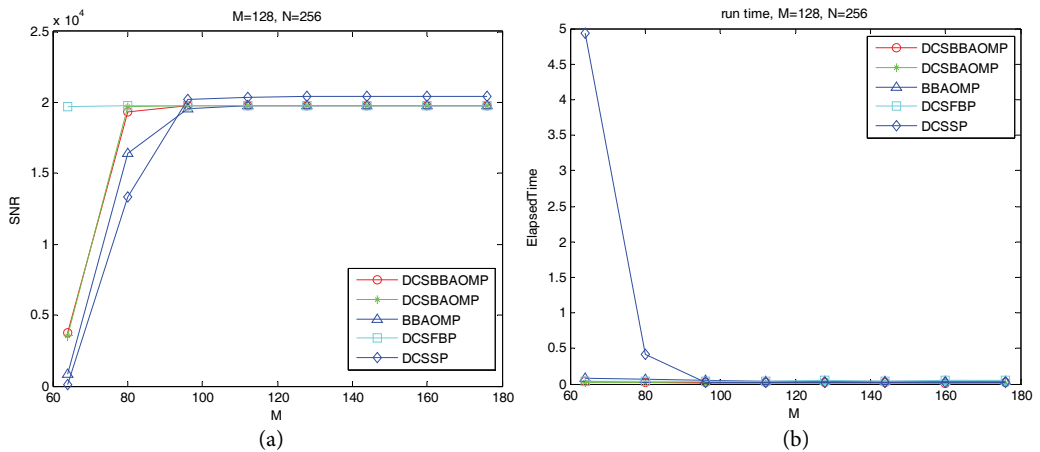


Fig. 2. Reconstruction results over the number of measurement M . The numerical values on x-axis denote the number of measurement M and those on y-axis represent the SNR (a) and run time (b).

3.2 The Recovery Performance versus Number of Measurement

In this experiment, we compare the SNR and run time as the number of measurement M varies from 64 to 176, where $N=256$, $\text{sparse_value}=30$, $J=3$.

From Fig. 2, as the number of measurement M increases, for all the five algorithms, the SNR increases while the run time decreases. Meanwhile, all the algorithms can obtain similar SNR and the run time with $M > 90$. When the number of measurement $M < 90$, performance of our algorithm is better than that of other algorithms except for DCSFBP. Although DCSSP has the largest SNR in this experiment, the run time of DCSSP is not stable compared with other algorithms.

3.3 The Recovery Performance versus Sparsity with Medium Sample

In this experiment, the recovery performance is observed as the sparse_value varies from 40 to 200 with the fixed block size $d=8$ and $M=512$, $N=1024$, $J=3$. The average SNR and run time are shown in the Fig. 3.

As can be seen from Fig. 3, the SNR curves decrease slightly and the run time curves increase as the sparse_value increases. It is obviously that DCSSP algorithm achieves the highest SNR, while the SNR of another four algorithms is more or less similar. However, the run time of DCSSP algorithm is longest among the above methods. To be mentioned, although the SNR of our method is not the highest, our method is more computational effective than other methods.

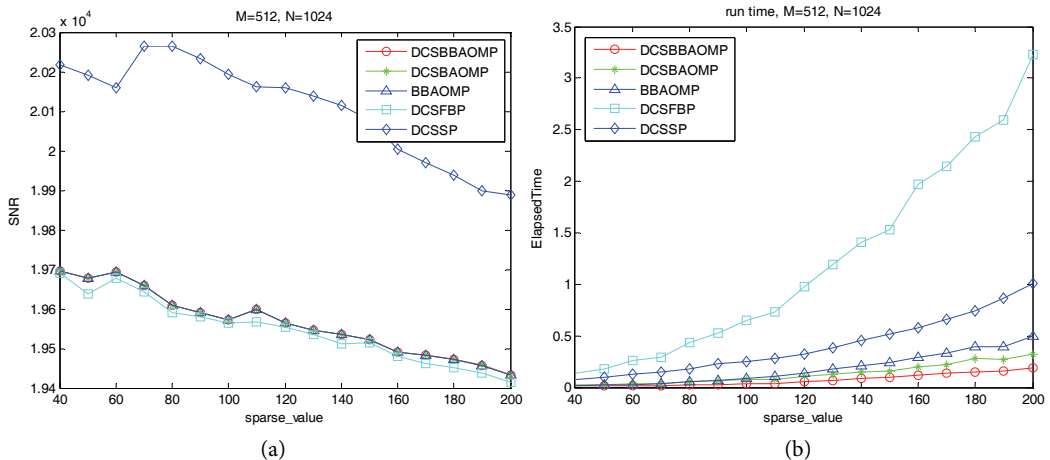


Fig. 3. Reconstruction results over the sparse_value . The numerical values on x-axis denote the sparse_value and those on y-axis represent the SNR (a) and run time (b).

3.4 The Recovery Performance versus Number of Block Size

In this experiment, we compare the SNR and run time as the number of block d varies from 4 to 16 with step size 4. $M=512$, $N=1024$, $\text{sparse_value}=120$, $J=3$ are fixed. Fig. 4 demonstrates the experimental results.

From Fig. 4, we can see that the SNR and run time of all the algorithms do not change any more as the block size d varies from 4 to 16. That is to say, if we fix the total sparsity of the block sparse signals, different block size of sparse signals does not affect the property of all the algorithms.

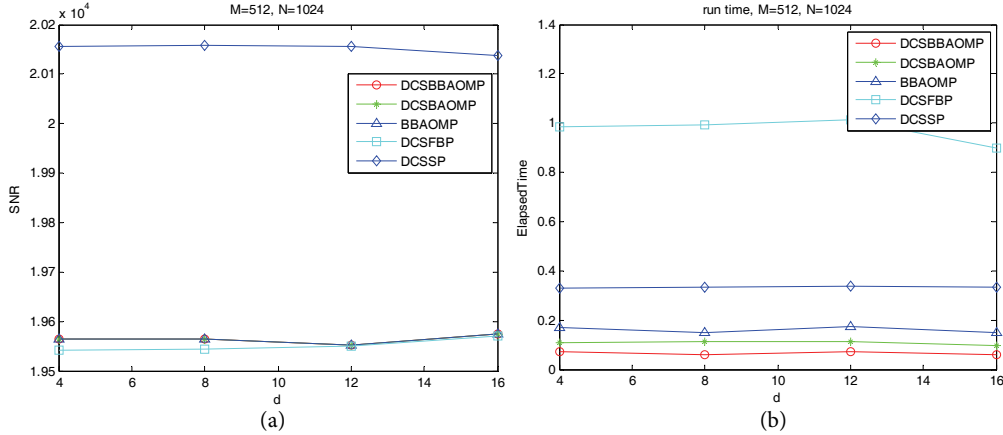


Fig. 4. Reconstruction results over the number of block d . The numerical values on x-axis denote the number of block d and those on y-axis represent the SNR (a) and run time (b).

3.5 The Recovery Performance with Unknown Block Size

3.5.1 The block size of our algorithm is fixed to 8

In this trail, we test the performance of our algorithm when the block size d is unknown. $M=512$, $N=1024$, $\text{sparse_value}=120$, $J=3$ are fixed. We generate the sources with block size $d=8$. Note that the block size is unknown in recovery process. Then we change the block size from 2 to 26 with step size 2 in the recovery process. The results are shown in Fig. 5.

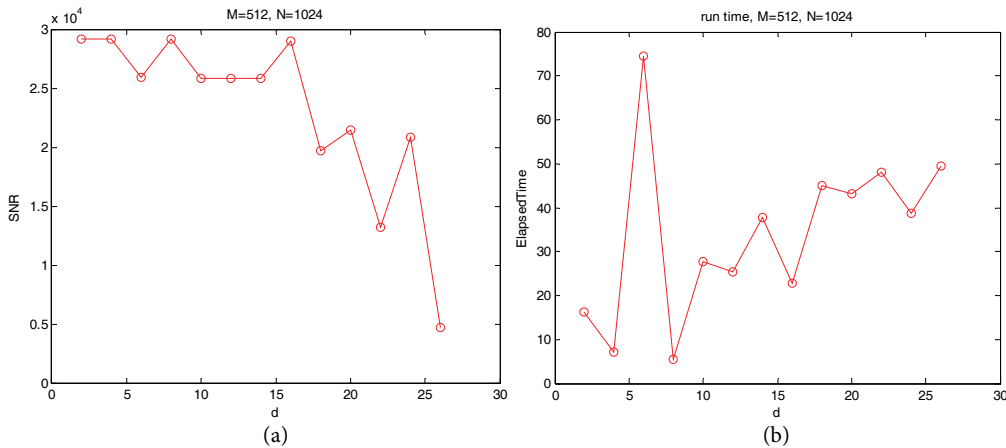


Fig. 5. Reconstruction results with unknown block d . The numerical values on x-axis denote the number of block d and those on y-axis represent the SNR (a) and run time (b); when we generate the source with block size $d=8$.

3.5.2 The block size of our algorithm is fixed to 5

In this trail, we generate the sources with block size $d=5$ instead of $d=8$. $M=512$, $N=1024$, $K_{\text{total}}=120$, $J=3$ are fixed. Then we change the block size from 3 to 16 with step size 1 in the recovery process. The results are shown in Fig. 6.

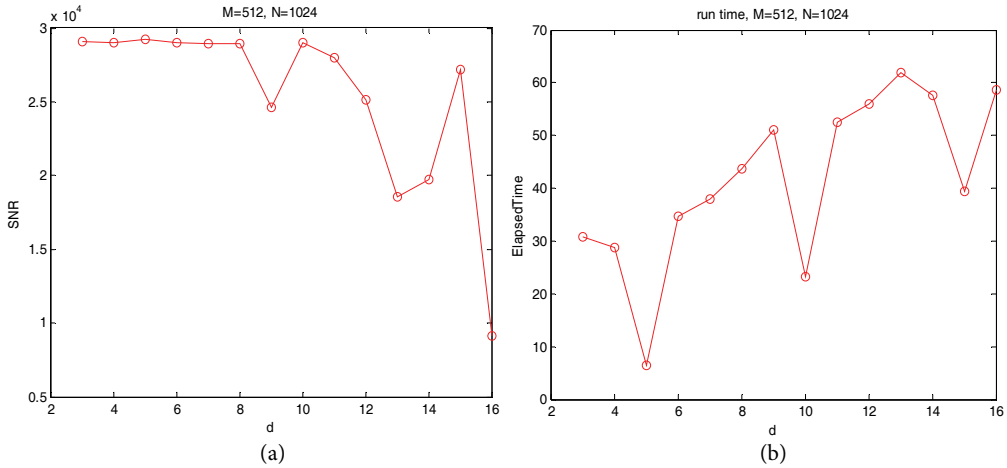


Fig. 6. Reconstruction results with unknown block d . The numerical values on x-axis denote the number of block d and those on y-axis represent the SNR (a) and run time (b); when we generate the source with block size $d = 5$.

From Figs. 5 and 6, when the block size choosing in our algorithm is the multiple of real block size, we can obtain much better reconstruction performance than the performance of other block size. Generally speaking, if the real block size d of sources is known in advance, we can get the best performance in the experiments with our algorithm.

3.6 ECG Signals Recovery

In this subsection, we apply our algorithm to the ECG data. Because the sparsity of signals need to be known in DCSSP, we only compare the performance of DCSBBAOMP with that of DCSBAOMP, DCSFBP, DCSSAMP and BBAOMP. We obtain the ECG data from the PhysioNet [29]. In this experiment, three patients are chosen randomly as the source signals \mathbf{X} . Then we randomly generate one Gaussian matrix $\Phi \in \mathbb{R}^{M \times N}$, where $N=1024, M=512$. The process of signals generation is the same as the literature [9]. From Fig. 7(a), we can see that the ECG data themselves are not sparse. Then we apply the orthogonal Daubechies wavelets (db1) Ψ to ECG signals. Fig. 7(b) shows the transformed sparse signals. Then all of the algorithms are applied to recover θ from $\mathbf{Y} = \Phi\mathbf{X} = \Phi\Psi\theta$. Fig. 7(c) depicts the recovered signals $\tilde{\mathbf{X}}$ and Fig. 7(d) shows the corresponding sparse signals $\tilde{\theta}$. The performance and the run time are shown in Table 1.

Table 1. Average reconstruction SNR and run time of block sparse signals using different methods

	Our algorithm	DCSBAOMP	DCSFBP	DCSSAMP	BBAOMP
Average SNR	180.2198	73.4399	69.5612	148.0398	200.0362
Run time (s)	20.6002	13.8757	51.2670	11.6684	78.2678

From Table 1, one can see that the performance of BBAOMP is best but it needs more time to recover the signals since this method is a single-channel method. Except for BBAOMP, our algorithm obtains the highest average SNR, and the run time is similar to other methods.

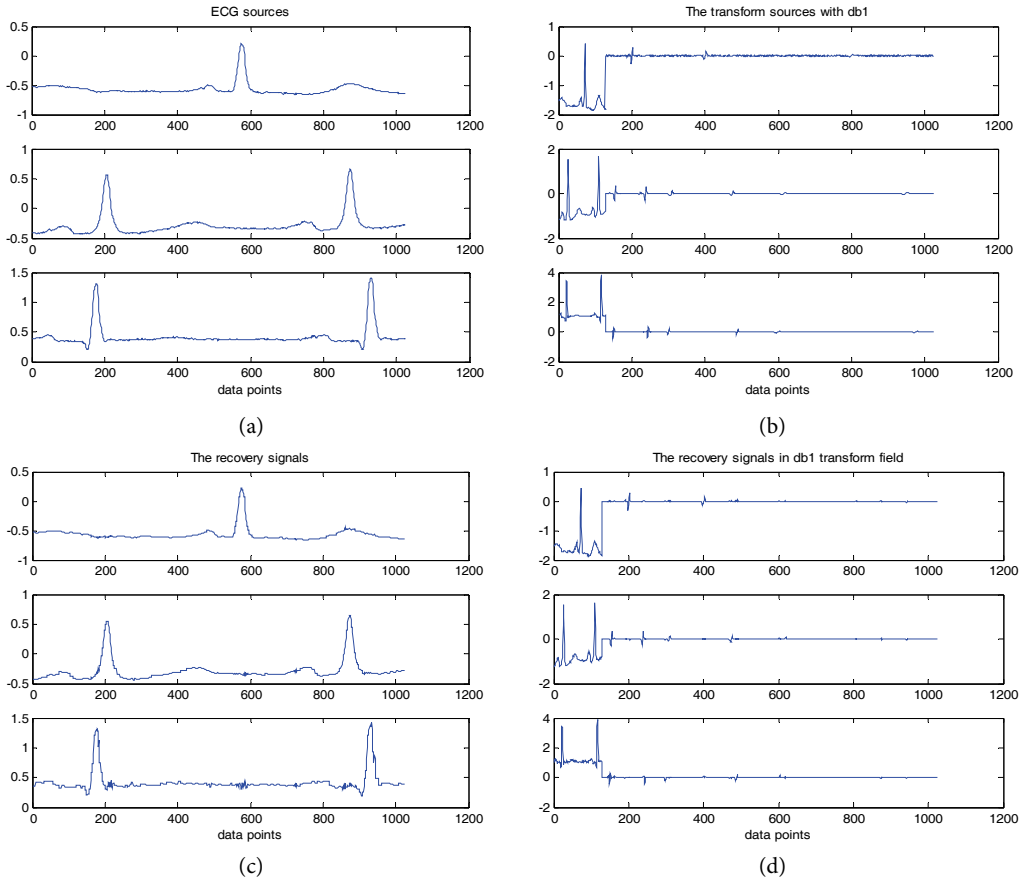


Fig. 7. The electrocardiography (ECG) signals in signal channel no#1 of three patients which are selected randomly from the PTB Diagnostic ECG Database: (a) the original signals \mathbf{X} , (b) $\boldsymbol{\theta}$ with orthogonal Daubechies waveles (db1), (c) $\tilde{\mathbf{X}}$ recovered by our algorithm, and (D) $\tilde{\boldsymbol{\theta}}$ recovered by our algorithm.

4. Conclusion

In this paper, a DCSBBAOMP method for recovery of block sparse signals is proposed. This method first chooses atoms adaptively and then removes some atoms that are wrongly chosen at the previous step by using backtracking procedure, which promotes the reconstruction property. The most useful advantage of our proposed algorithm is that it can recover multiple sparse signals from their compressive measurements simultaneously. What's more, it does not need the block sparsity as a prior. Simulation results demonstrate that our method produces much better reconstruction property compared with many existing algorithms.

The two parameters μ_1 and μ_2 play a key role in our method which provides some flexibility between reconstruction property and computational complexity. However, there is no theoretical guidance on how to select μ_1 and μ_2 . In addition, theoretical guarantees of that the proposed method can accurately recovery the original signals are also not proved. Future works include theoretical analysis of exact reconstruction of the proposed algorithm and the treatment of the selection of parameters of μ_1 and μ_2 .

Acknowledgement

This work is supported by Natural Science Foundation of China (No. 61601417).

References

- [1] Q. Wang and Z. Liu, "A robust and efficient algorithm for distributed compressed sensing," *Computers & Electrical Engineering*, vol. 37, no. 6, pp. 916-926, 2011.
- [2] H. Palangi, R. Ward, and L. Deng, "Convolutional deep stacking networks for distributed compressive sensing," *Signal Processing*, vol. 131, pp. 181-189, 2017.
- [3] Y. Oktar and M. Turkan, "A review of sparsity-based clustering methods," *Signal Processing*, vol. 148, pp. 20-30, 2018.
- [4] L. Vidya, V. Vivekanand, U. Shyamkumar, and M. Deepak, "RBF network based sparse signal recovery algorithm for compressed sensing reconstruction," *Neural Networks*, vol. 63, pp. 66-78, 2015.
- [5] X. Li, H. Bai, and B. Hou, "A gradient-based approach to optimization of compressed sensing systems," *Signal Processing*, vol. 139, pp. 49-61, 2017.
- [6] G. Coluccia, A. Roumy, and E. Magli, "Operational rate-distortion performance of single-source and distributed compressed sensing," *IEEE Transactions on Communications*, vol. 62, no. 6, pp. 2022-2033, 2014.
- [7] D. Baron, M. F. Duarte, M. B. Wakin, S. Sarvotham, and R. G. Baraniuk, "Distributed compressed sensing," 2009 [Online]. Available: <https://arxiv.org/abs/0901.3403>.
- [8] Y. J. Zhang, R. Qi, and Y. Zeng, "Backtracking-based matching pursuit method for distributed compressed sensing," *Multimedia Tools and Applications*, vol. 76, no. 13, pp. 14691-14710, 2017.
- [9] Y. Zhang, R. Qi, and Y. Zeng, "Forward-backward pursuit method for distributed compressed sensing," *Multimedia Tools and Applications*, vol. 76, no. 20, pp. 20587-20608, 2017.
- [10] Y. C. Eldar and H. Rauhut, "Average case analysis of multichannel sparse recovery using convex relaxation," *IEEE Transactions on Information Theory*, vol. 56, no. 1, pp. 505-519, 2010.
- [11] D. L. Donoho, "For most large underdetermined systems of linear equations the minimal l_1 -norm solution is also the sparsest solution," *Communications on Pure and Applied Mathematics*, vol. 59, no. 6, pp. 797-829, 2006.
- [12] M. F. Duarte, S. Sarvotham, D. Baron, M. B. Wakin, and R. G. Baraniuk, "Distributed compressed sensing of jointly sparse signals," in *Proceeding of the 39th Asilomar Conference on Signals, Systems and Computers*, Pacific Grove, CA, 2005, pp. 1537-1541.
- [13] M. B. Wakin, M. F. Duarte, S. Sarvotham, D. Baron, and R. G. Baraniuk, "Recovery of jointly sparse signals from few random projections," *Advances in Neural Information Processing Systems*, vol. 18, pp. 1435-1440, 2005.
- [14] J. A. Tropp, A. C. Gilbert, and M. Strauss, "Simultaneous sparse approximation via greedy pursuit," in *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, Philadelphia, PA, 2005.
- [15] D. Sundman, S. Chatterjee, and M. Skoglund, "Greedy pursuits for compressed sensing of jointly sparse signals," in *Proceedings of 2011 19th European Signal Processing Conference*, Barcelona, Spain, 2011, pp. 368-372.
- [16] K. Lee, Y. Bresler, and M. Junge, "Subspace methods for joint sparse recovery," *IEEE Transactions on Information Theory*, vol. 58, no. 6, pp. 3613-3641, 2012.
- [17] X. T. Yuan, X. Liu, and S. Yan, "Visual classification with multitask joint sparse representation," *IEEE Transactions on Image Processing*, vol. 21, no. 10, pp. 4349-4360, 2012.
- [18] F. Parvaresh, H. Vikalo, S. Misra, and B. Hassibi, "Recovering sparse signals using sparse measurement matrices in compressed DNA microarrays," *IEEE Journal of Selected Topics in Signal Processing*, vol. 2, no. 3, pp. 275-285, 2008.

- [19] S. F. Cotter and B. D. Rao, "Sparse channel estimation via matching pursuit with application to equalization," *IEEE Transactions of Communications*, vol. 50, no. 3, pp. 374-377, 2002.
- [20] R. Qi, D. Yang, Y. Zhang, and H. Li, "On recovery of block sparse signals via block generalized orthogonal matching pursuit," *Signal Processing*, vol. 153, pp. 34-46, 2018.
- [21] R. G. Baraniuk, V. Cevher, M. F. Duarte, and C. Hegde, "Model-based compressive sensing," *IEEE Transactions on Information Theory*, vol. 56, no. 4, pp. 1982-2001, 2010.
- [22] D. Needell and J. A. Tropp, "CoSaMP: iterative signal recovery from incomplete and inaccurate samples," *Applied and Computational Harmonic Analysis*, vol. 26, no. 3, pp. 301-321, 2009.
- [23] Y. C. Eldar, P. Kuppinger, and H. Bolcskei, "Block-sparse signals: Uncertainty relations and efficient recovery," *IEEE Transactions on Signal Processing*, vol. 58, no. 6, pp. 3042-3054, 2010.
- [24] L. Zelnik-Manor, K. Rosenblum, and Y. C. Eldar, "Dictionary optimization for block-sparse representations," *IEEE Transactions on Signal Processing*, vol. 60, no. 5, pp. 2386-2395, 2012.
- [25] A. Kamali, M. A. Sahaf, A. D. Hooseini, and A. A. Tadaion, "Block subspace pursuit for block-sparse signal reconstruction," *Iranian Journal of Science and Technology: Transactions of Electrical Engineering*, vol. 37, no. E1, pp. 1-16, 2013.
- [26] B. X. Huang and T. Zhou, "Recovery of block sparse signals by a block version of StOMP," *Signal Processing*, vol. 106, pp. 231-244, 2015.
- [27] R. Qi, Y. Zhang, and H. Li, "Block sparse signals recovery via block backtracking-based matching pursuit method," *Journal of Information Processing Systems*, vol. 13, no. 2, pp. 360-369, 2017.
- [28] E. Candes, J. Romberg, and T. Tao, "Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information," *IEEE Transactions on Information Theory*, vol. 52, no. 2, pp. 489-509, 2006.
- [29] A. L. Goldberger, L. A. Amaral, L. Glass, J. M. Hausdorff, P. C. Ivanov, R. G. Mark, J. E. Mietus, G. B. Moody, C. K. Peng, and H. E. Stanley, "PhysioBank, PhysioToolkit, and PhysioNet: components of a new research resource for complex physiologic signals," *Circulation*, vol. 101, no. 23, pp. e215-e220, 2000.



Xingyi Chen <https://orcid.org/0000-0003-1943-3925>

He graduated from the School of Geodesy and Geomatics, Wuhan University, in 1984. He is currently a professor of China University of Geosciences information Engineering College, China. His research interests include photogrammetry and remote sensing.



Yujie Zhang <https://orcid.org/0000-0001-7710-4017>

She received the M.S. degree in applied mathematics and Ph.D. degree in Institute of Geophysics and Geomatics from China University of Geosciences, China, in 2006 and 2012, respectively. She is currently a lecturer at the China University of Geosciences, China. Her research interests include blind signal processing, time-frequency analysis and their applications.



Rui Qi <https://orcid.org/0000-0003-3183-2427>

He received the M.S. degree in School of Mathematics and Statistics from Huazhong University of Science and Technology, China, in 2009. He is now a PhD candidate of the Institute of Geophysics and Geomatics of China University of Geosciences, China. His research interests include sparse representation and compressed sensing.