

Research Article

A New Generalized Orthogonal Matching Pursuit Method

Liquan Zhao and Yulong Liu

College of Information Engineering, Northeast Electric Power University, Jilin 132012, China

Correspondence should be addressed to Liquan Zhao; zhao.liquan@163.com

Received 26 January 2017; Revised 1 July 2017; Accepted 16 July 2017; Published 13 August 2017

Academic Editor: Jar Ferr Yang

Copyright © 2017 Liquan Zhao and Yulong Liu. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

To improve the reconstruction performance of the generalized orthogonal matching pursuit, an improved method is proposed. Columns are selected from the sensing matrix by generalized orthogonal matching pursuit, and indices of the columns are added to the estimated support set to reconstruct a sparse signal. Those columns contain error columns that can reduce the reconstruction performance. Therefore, the proposed algorithm adds a backtracking process to remove the low-reliability columns from the selected column set. For any k -sparse signal, the proposed method firstly computes the correlation between the columns of the sensing matrix and the residual vector and then selects s columns that correspond to the s largest correlation in magnitude and adds their indices to the estimated support set in each iteration. Secondly, the proposed algorithm projects the measurements onto the space that consists of those selected columns and calculates the projection coefficient vector. When the size of the support set is larger than k , the proposed method will select k high-reliability indices using a search strategy from the support set. Finally, the proposed method updates the estimated support set using the selected k high-reliability indices. The simulation results demonstrate that the proposed algorithm has a better recovery performance.

1. Introduction

In recent years, a new theory named compressive sensing (CS) [1] has surpassed the limits of the Nyquist sampling rate. Because CS can recover signals at a sampling frequency far lower than the Nyquist sampling rate, CS has aroused tremendous interests over the past few years [2, 3]. CS differs from the traditional Nyquist sampling theory and includes three procedures: sparse representation, nonrelated linear measurement, and signal reconstruction. The reconstruction algorithm aims to recover signals accurately from the measurements, and this step is one of most important parts of CS.

Recently, many reconstruction algorithms have been proposed to obtain the original sparse signal from measurements. Two major classes of reconstruction algorithms are l_1 -minimization and greedy pursuit algorithms. Common l_1 -minimization approaches include basis pursuit (BP) [4], Gradient projection for sparse reconstruction (GPSR) [5], iterative thresholding (IT) [6], and other algorithms. Those algorithms possess good performance in solving a convex minimization problem, but they have a higher computational complexity.

Greedy algorithms have received increasing attention for their excellent performance and small cost in recovering sparse signals from compressed measurements. A greedy algorithm proposed early on was the matching pursuit algorithm (MP) [7]. Building on the MP algorithm, the orthogonal matching pursuit algorithm (OMP) [8] was proposed to optimize the MP via orthogonalization of the estimate support set. The OMP has become a well-known greedy algorithm with wide application. The regularized orthogonal matching pursuit algorithm (ROMP) [9] was developed to refine the selected columns of the measurement matrix with a regularized rule to improve the speed of OMP. The stage wise orthogonal matching pursuit (StOMP) [10] selects multiple columns in each iteration via a presupposed threshold. The subspace pursuit (SP) [11] and compressive sampling matching pursuit (CoSaMP) [12] proposed similar improvement methods. Both of these algorithms were proposed with the idea of backtracking, and the difference is that SP selects k columns from the sensing matrix for each iteration, while CoSaMP selects $2k$. The generalized orthogonal matching pursuit (GOMP) was proposed by Wang et al. [13, 14]. The algorithm selects S ($S \leq K$) columns in each iteration.

When $S = 1$, GOMP is identical to OMP. Compared to OMP, which selects only one column in each iteration, GOMP changes the number of columns that are selected in each iteration to improve the computational efficiency and recovery performance. The generalized OMP (GOMP) has received increasing attention in recent years, because the method can enhance the recovery performance of OMP. Several papers have been published on the analysis of the theoretical performance of GOMP [13–17].

2. Compressive Sensing Model

Compressive sensing requires that the target signal is a K -sparse signal. It means that if we regard the signal as a dimensional vector x , there should be at most K no-zero elements in x . However, in practical applications, sparse signals may not exist in all cases. The target signal has to be transformed into a sparse signal based on a set of sparse basis $\Psi = \{\phi_1, \phi_2, \phi_3, \dots, \phi_N\}$. In this case, x can be defined as

$$x = \sum_{i=1}^N \alpha_i \phi_i = \Psi \alpha, \quad (1)$$

where $\|\alpha\|_0 = K$; $\|\cdot\|_0$ denotes the number of nonzero elements in a vector. Thus, the signal x is equivalently represented by K -sparse vector α under some linear transformation Ψ in some domains. The process of compressive sensing can be regarded as a technique that automatically selects relevant information from signals by a measurement. In the theory, x is translated into M -dimensional measurements y via a matrix multiplication with Φ . We express it as

$$y = \Phi x, \quad (2)$$

where Φ is defined as the measurement matrix with dimensions $M \times N$. Combining (1) with (2), we can obtain

$$y = \Phi x = \Phi \Psi \alpha = A_{CS} \alpha, \quad (3)$$

where $A_{CS} = \Phi \Psi$.

In most scenarios, $M \ll N$. Thus, we can interpret that x is compressed into y with a dimension ranging from N to M . Clearly, (3) is an underdetermined equation, and it is difficult to obtain an accurate solution based on the equation. That is to say, it is impossible with traditional methods that obtain the inverse of the matrix Φ to reconstruct the original signal x accurately. In this case, we can obtain x by solving the l_p -minimization problem:

$$\begin{aligned} \hat{\alpha} = \min \quad & \|\alpha\|_p \\ \text{s.t.} \quad & y = A_{CS} \alpha. \end{aligned} \quad (4)$$

Several methods exist for solving this problem. When $p = 1$, the problem is a l_1 -minimization problem, which can be solved by using a convex optimization algorithm. When $p = 0$, the problem is a l_0 -minimization problem, which can be solved using a greedy algorithm. An appropriate condition for exact recovery is that the matrix A_{CS} satisfies the condition of restricted isometry property (RIP) condition [1].

Definition 1. A sensing Matrix A_{CS} is said to satisfy the RIP condition with the smallest number of the K -restricted isometry constant δ_K ($\delta_K \in (0, 1)$), if

$$(1 - \delta_K) \|\alpha\|^2 \leq \|A_{CS} \alpha\|^2 \leq (1 + \delta_K) \|\alpha\|^2 \quad (5)$$

holds for any K -sparse vector $\alpha \in R^{N \times 1}$ with $\|\alpha\|_0 \leq K$.

3. GOMP Algorithm

Greedy algorithms are used widely to recover signals in CS due to their simple algorithms and low computational complexity. Two methods exist for improving the OMP algorithm. The first method is based on the idea of backtracking, as in the subspace pursuit algorithm (SP) [11]. The second method selects more than one atom in each iteration, as in GOMP. The computational complexity of the backtracking method is higher, but it yields higher accuracy in most cases. In this section, we will introduce the GOMP algorithm.

GOMP firstly computes the correlation between the columns of the sensing matrix Φ and the residual vector r^{k-1} by $\Phi' r^{k-1}$ and then selects S columns that correspond to the S largest correlation in magnitude adding their indices to the estimated support set Λ^k in each iteration. Next the projection coefficient vector \hat{x}_{Λ^k} of measurements y onto space of $\text{span}(\Phi_{\Lambda^k})$ is obtained using the least square method (LS). The residual r^k is revised by subtracting $\Phi_{\Lambda^k} \hat{x}_{\Lambda^k}$ from y . These operations are repeated until either the iteration number reaches the maximum $k = \min\{K, M/S\}$, or the l_2 -norm of the residual falls below a threshold ϵ . For ease of understanding, we describe the GOMP algorithm in Algorithm 1 according to [13, 14].

In Algorithm 1, we observe that the only difference between GOMP and OMP is that the GOMP selects more than one atom in each iteration. The GOMP algorithm selects S ($S \geq 1$) atoms in each iteration. When $S = 1$, the GOMP is identical to OMP.

4. Improved GOMP Algorithm

For the usage of greedy algorithms, it is important to generate an estimate of the correct support set. We assume the correct support is T , and the estimate of T is Λ^k . The goal is to determine the indices for Λ^k that are most similar to T . GOMP selects S columns from the sensing matrix Φ and added the indices of these columns to the estimated support set Λ^k to reconstruct a sparse signal. Those columns contain columns that be selected by error. Although GOMP allows for a number of error indices in Λ^k , these error indices will lead to the size of Λ^k larger than K . This increases the algorithm complexity greatly in the process of estimation of x_{Λ^k} and residual update in the GOMP.

In the process of estimation of x_{Λ^k} , the projection coefficient vector of the measurements onto the space of $\text{span}(\Phi_{\Lambda^k})$ can be expressed as

$$\hat{x}_{\Lambda^k} = \Phi_{\Lambda^k}^\dagger y, \quad (6)$$

```

Input:   measurements  $y \in R^{M \times N}$ ,
         sensing matrix  $\Phi \in R^{M \times N}$ ,
         Sparsity  $K$ ,
Initialize: number of indices of columns for each
selection  $S$  ( $S \leq K$ ).
         iteration count  $k = 0$ ,
         residual vector  $r^0 = y$ ,
         estimated support set  $\Lambda^0 = \emptyset$ .
While    $\|r^k\| > \varepsilon$  and  $k < \min\{K, M/S\}$  do
          $k = k + 1$ ;
         (Identification)
         Select  $S$  largest entries (in magnitude) from  $\Phi' r^{k-1}$ .
         Then record the indices  $\{\varphi(i)\}_{i=1,2,3,\dots,S}$  corresponding
         to the entries.
         (Augmentation)
          $\Lambda^k = \Lambda^{k-1} \cup \{\varphi(1), \varphi(2), \dots, \varphi(S)\}$ .
         (Estimation of  $x_{\Lambda^k}$ )
          $\hat{x}_{\Lambda^k} = \arg \min_{\sup p(u)=\Lambda^k} \|y - \Phi u\|_2$ .
         (Residual Update)
          $r^k = y - \Phi_{\Lambda^k} \hat{x}_{\Lambda^k}$ .
End
Output The estimated support  $\hat{\Lambda} = \arg \min_{T: \|T\|=K} \|\hat{x}_{\Lambda^k} - \hat{x}_T\|_2$ 
and signal  $\hat{x}_{\hat{\Lambda}} = \Phi_{\hat{\Lambda}}^\dagger y$ .

```

ALGORITHM 1: GOMP algorithm.

where $\Phi_{\Lambda^k}^\dagger = (\Phi_{\Lambda^k}^T \Phi_{\Lambda^k})^{-1} \Phi_{\Lambda^k}^T$; y are the measurements of the signal vector x . The cost of (6) is

$$4S^2 km + (-2S^2 + 5N)m + 2S^3 k^2 + (-4S^3 + 5S^2)k + 3S^3 - S^2 - S, \quad (7)$$

where S represents the number of indices of columns for each selection, k is the number of iterations, and m is the dimensionality of measurements y [13].

In the process of estimation of residual update, the residuals in GOMP can be expressed as

$$r^k = y - \Phi_{\Lambda^k} \hat{x}_{\Lambda^k}, \quad (8)$$

where y represents the measurements, Φ_{Λ^k} is the estimated support set, and \hat{x}_{Λ^k} is the projection coefficient vector of the measurements onto the space of the estimated support set. The cost of $\Phi_{\Lambda^k} \hat{x}_{\Lambda^k}$ is

$$(2kS - 1)m, \quad (9)$$

where k is the number of iterations and S is the number of indices of columns for each selection.

GOMP selects S indices in each iteration and adds them to Λ^k . On the k -th iteration, kS indices will be selected, and the dimensionality of Φ_{Λ^k} is $m \times kS$. If all the indices in the estimated support set Λ^k are correct, Λ^k is identical to T and $kS \approx K$. Although a selection rule exists to ensure that newly added indices belong to the correct support T in GOMP, it is unavoidable that the error indices are selected and once selected, the error indices of the selected columns

will remain in the support set throughout the remainder of the reconstruction process. It means $kS > K$. Generally speaking, when a large S is selected, the GOMP should exert higher efficiency than a small S . However, as S increases, the probability of the selected error indices increases as well. When a large number of error indices are selected, it leads to a great increase in kS , further increasing the cost of the algorithm.

To overcome this problem, we propose a method to improve the performance of GOMP. The proposed method will retain the size of the estimated support set as K and update the indices using the backtracking method to reduce the number of error indices, when the number of indices is greater than K . Even if an index is deemed reliable in some iteration, when it is considered unreliable in subsequent iterations, the index will be removed from the estimated support set. This can reduce the number of error indices in the estimated support set.

The main difference between the proposed and the conventional GOMP algorithm is that the new algorithm can add or remove the index from the estimated support set at any stage of the recovery process. The proposed algorithm increases the reliability of the indices of the estimated support set and furthermore reduces the cost of the reconstruction process. The process of proposed algorithm is expressed in Algorithm 2.

In Algorithm 2, we observe that the proposed method adds a backtracking step to the GOMP algorithm. When S is small, it has a high accuracy for the identification process. When $S = 1$, GOMP is identical to OMP. At this step, GOMP has the highest accuracy in the identification step.

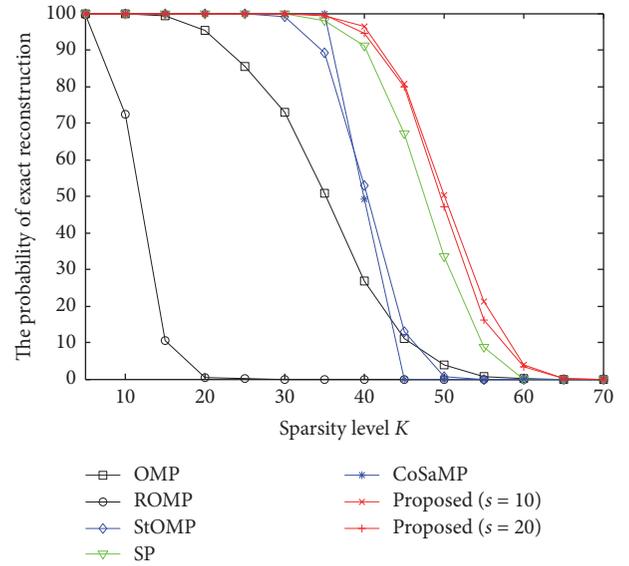
Input: measurements $y \in R^{M \times N}$,
sensing matrix $\Phi \in R^{M \times N}$,
Sparsity K ,
Initialize: number of indices of columns for each selection
 S ($S \leq K$).
iteration count $k = 0$,
residual vector $r^0 = y$,
estimated support set $\Lambda^0 = \emptyset$.
While $\|r^k\| > \varepsilon$ and $k < \min\{K, M/S\}$ do
 $k = k + 1$;
(Identification)
Select S largest entries (in magnitude) from $\Phi' r^{k-1}$. Then
record the indices $\{\varphi(i)\}_{i=1,2,3,\dots,S}$ corresponding to the
entries.
(Augmentation)
 $\Lambda^k = \Lambda^{k-1} \cup \{\varphi(1), \varphi(2), \dots, \varphi(S)\}$.
(Estimation of x_{Λ^k})
 $x_{\Lambda^k} = \arg \min_{\text{supp } p(u)=\Lambda^k} \|y - \Phi u\|_2$.
(Backtracking) While $\|x_{\Lambda^k}\|_0 \geq K$, select K largest
elements of $|x_{\Lambda^k}|$. Then recording the indices corresponding
to the elements, and renew the Λ_k with those indices.
(Residual Update) $r^k = y - \Phi_{\Lambda^k} \hat{x}_{\Lambda^k}$.
End
Output The estimated support $\hat{\Lambda} = \arg \min_{T: \|T\|=K} \|\hat{x}_{\Lambda^k} - \hat{x}_T\|_2$
and signal $\hat{x}_{\hat{\Lambda}} = \Phi_{\hat{\Lambda}}^\dagger y$.

ALGORITHM 2: Improved method of GOMP.

As S increases, the probability of the selected error indices increases as well. The proposed method adds a backtracking process to remove those error indices in subsequent iterations. When $S \leq K$, the proposed method requires more than one iteration to create a K -sized support set. When $kS < K$, the support set cannot contain K correct indices, and executing the backtracking step is unnecessary and will cause wasteful computation. Therefore, we designed the algorithm to execute the backtracking step when $kS \geq K$. The dimensionality of x_{Λ^k} is $kS \times 1$ before executing the backtracking step. So the improved algorithm determines whether to execute the backtracking step by the size of x_{Λ^k} . Experimental evidence demonstrated that our changes improved the performance. We will describe our simulation in the next section.

5. Simulation and Discussion

In this part, we will demonstrate the performance of the proposed algorithm with sparse signals. We used the same sparse signal sources as in GOMP with K -sparse to compare the performances of the different algorithms. The components of the sensing matrix were generated randomly with Gaussian distribution, and the size of the sensing matrix was 128×256 . We used MATLAB 7.0 with a quad-core 64-bit processor in a Windows 10 environment. We executed each algorithm 1000 times and recorded the probability of the exact reconstructions. We set the threshold $\varepsilon = 10^{-6}$ used in [13] in both GOMP and the proposed method.

FIGURE 1: Reconstruction performance for K -sparse Gaussian signal vector as a function of sparsity K .

In Figure 1, we compare the probability of successful reconstruction of the proposed method with the ROMP [9], OMP [8], SP [11], CoSaMP [12], and StOMP [10] algorithms. It is evident from Figure 1 that the reconstruction performance of ROMP decreased rapidly with a rise in K .

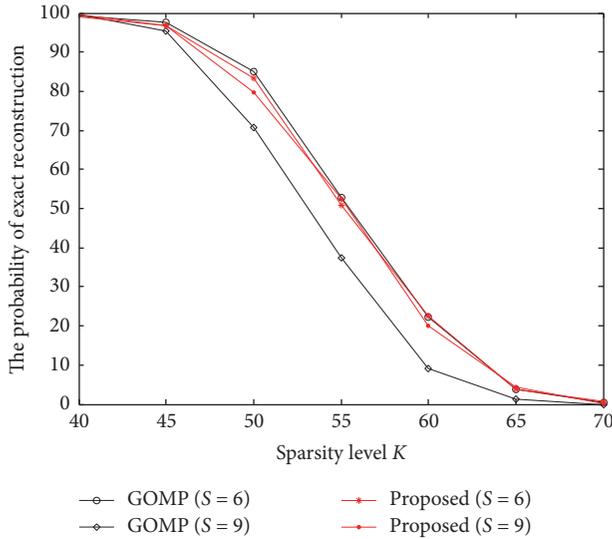


FIGURE 2: Reconstruction performance of GOMP and the proposed method with a small S value.

When $K = 20$, the recovery probability was near zero. The probability of successful recovery of OMP decreased at a smaller rate than that of ROMP, and its critical sparsity was reached at $K = 55$. CoSaMP and StOMP had similar performances. Both retained a high recovery performance when $K \leq 40$. The probability of successful recovery of SP was higher than for ROMP, OMP, CoSaMP, and StOMP with the same K . For our proposed method, we tested the performance with different S values, representing the number of selected columns in each iteration. We set $S = 10$ and $S = 20$. Figure 1 indicates that the smaller the value of S , the higher the probability of successful recovery of the proposed algorithm with the same K . Compared to the other algorithms, under the same conditions, the proposed method with $S = 10$ had the highest probability of successful recovery, followed by the proposed method with $S = 20$, and followed by the SP algorithm. The proposed algorithm with different S values had a higher probability than the other algorithms.

In Figures 2 and 3, we compare the performances of the proposed method and GOMP with different values for the parameter S . In Figure 2, we selected smaller S values to compare the algorithms. All algorithms resulted in high levels of probability of recovery when $K \leq 40$. The probability of exact recovery decreased with an increase in K and reached zero when $K = 70$. Both the proposed algorithm and GOMP had a higher probability of exact recovery with a smaller S value. When $S = 6$, the two algorithms had nearly the same probability values. When $S = 9$, the proposed method had a higher probability of recovery. The probability of exact reconstruction of both algorithms was 100% when $K < 40$ and zero when $K > 70$ for $S = 6$ and $S = 9$, and the difference between two algorithms mainly concentrated on the sparsity level from 40 to 70. Therefore, we compared two algorithms with $K = 40$ to $K = 70$ in Figure 2.

In Figure 3, we selected a larger value of S to compare the proposed algorithm and GOMP. The probability of exact

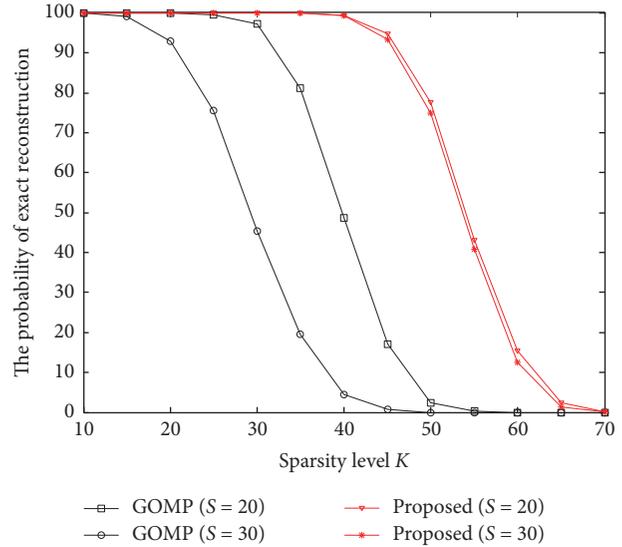


FIGURE 3: Reconstruction performance of GOMP and the proposed method with a large S value.

reconstruction of both algorithms was 100% for $K < 10$ and zero for $K > 70$, so we compared the two algorithms with $K = 10$ to $K = 70$ for $S = 20$ and $S = 30$ in Figure 3.

When $K = 40$ and $S = 20$, the probability of exact recovery of GOMP was 48.7%, while the probability value for the proposed method was 99.4%. When $K = 40$ and $S = 30$, the resulting probability values were 4.5% for GOMP and 99.4% for the proposed method. These results indicated that, for different S values, the proposed method had a higher probability of exact recovery than the GOMP method.

Figures 2 and 3 demonstrate that, for different S values, the curves of probability for exact recovery were closer to each other for the proposed method than the GOMP method. This indicated that the proposed method had a more stable performance than the GOMP method for different S values.

Based on the analysis and comparison, we determined that even though the proposed algorithm had an additional backtracking process compared to GOMP, the proposed method demonstrated an excellent performance with regard to running time. In order to compare running times for all the algorithms, we ran each algorithm one thousand times to calculate the average running time. The computing environment was the same as for the determination of the probability of exact reconstruction.

In Figure 2, we can see that the probability of exact reconstruction of both algorithms was below 100% for $K > 40$ with $S = 6$ and $S = 9$, and the effective running time should be computed for successful reconstructing. Therefore, we just compared the running time with $K < 40$ for two methods. The difference was not obvious with $K < 5$ in running time for both methods with $S = 6$ and $S = 9$, so we compared the running time with $K > 5$ and $K < 40$ in Figure 4.

In Figure 3, we can see that probability of exact reconstruction of GOMP algorithm with $S = 20$ was below 100% for $K < 20$, and with $S = 30$ for $K < 10$, and the probability of exact reconstruction of proposed algorithm with $S = 20$ and

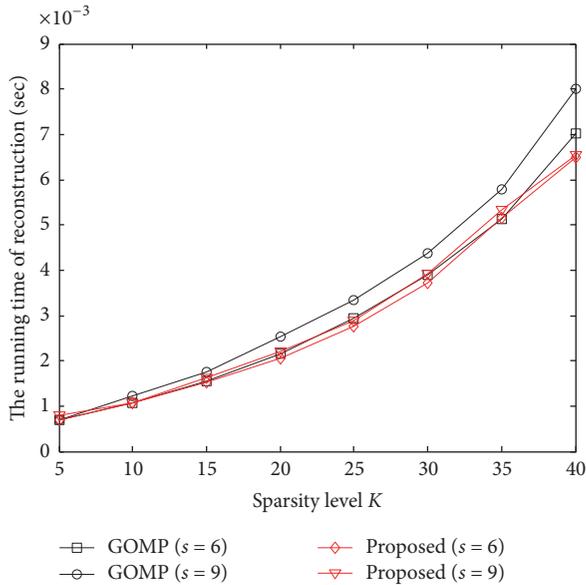


FIGURE 4: The average running time of GOMP and the proposed method with a small S value.

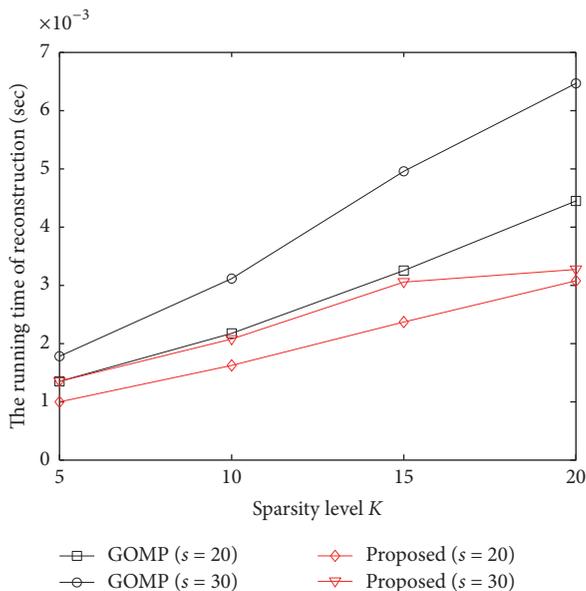


FIGURE 5: The average running time of GOMP and the proposed method with a large S value.

$S = 30$ all were below 100% for $K < 40$. The effective running time should be computed for successful reconstructing, so we just compared the running time of both algorithms for $S = 20$ with $K < 20$, and $S = 30$ with $K < 10$. The difference was not obvious in running time for both methods with smaller K . Based on the above analysis, we just compared their running time with $K > 5$ and $K < 20$ for $S = 20$, and $K > 5$ and $K < 10$ for $S = 30$. In order to unify the sparsity level scope, we set K from 5 to 20 in Figure 5.

In Figure 4, we set $S = 6$ and $S = 9$ to compare the running times for GOMP and the proposed method. The running

times for GOMP with $S = 6$ and the proposed method with $S = 6$ and $S = 9$ were nearly identical. The running times under those conditions were less than those for GOMP with $S = 9$. When $K = 40$, the running time of the proposed method with $S = 6$ and $S = 9$ was less than that of GOMP with $S = 6$.

In Figure 5, we set $S = 20$ and $S = 30$. From Figure 5, we can observe that the running time of the proposed algorithms was less than GOMP for $S = 20$ with $K = 5$ to 20, and for $S = 30$ with $K = 5$ to 10. This shows that the proposed algorithm has faster speed for reconstructing.

These results show that the proposed method resulted not only in better performance with regard to running time, but also in a higher probability of exact reconstruction. The advantage is more obvious when larger S is taken in both algorithms.

6. Conclusion

In this paper, a novel method for sparse signal reconstruction has been proposed. The proposed method adds a backtracking step to maintain a K -sized estimated support set, avoiding the extra computation cost for an oversized estimated set. At the same time, the proposed method can increase the reliability of the estimated support set by removing the low-reliability columns from the estimated support set. The proposed reconstruction algorithm performs well at not only reconstructing the sparse signal (i.e., when K is small), but also the less sparse signal (i.e., when K is large). The simulation results showed that the proposed method had a better performance than GOMP with regard to both the probability of exact reconstruction and running time, especially with larger S values. In future research, we want to improve on the proposed algorithm to optimize the greedy algorithm further.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (61271115) and the Foundation of Jilin Educational Committee (2015235).

References

- [1] D. L. Donoho, "Compressed sensing," *Institute of Electrical and Electronics Engineers. Transactions on Information Theory*, vol. 52, no. 4, pp. 1289–1306, 2006.
- [2] R. Zhao and H. N. Yu, "Compressed sensing, based over-complete dictionary learning algorithm for images," *Journal of Northeast Dianli University*, vol. 32, no. 4, pp. 44–47, 2012.
- [3] H. N. Yu and L. Li, "Compressed sensing," *Journal of Northeast Dianli University*, vol. 32, no. 1, pp. 79–83, 2011.
- [4] S. S. Chen, D. L. Donoho, and M. A. Saunders, "Atomic decomposition by basis pursuit," *SIAM Journal on Scientific Computing*, vol. 20, no. 1, pp. 33–61, 1998.
- [5] M. A. T. Figueiredo, R. D. Nowak, and S. J. Wright, "Gradient projection for sparse reconstruction: Application to compressed

- sensing and other inverse problems,” *IEEE Journal on Selected Topics in Signal Processing*, vol. 1, no. 4, pp. 586–597, 2007.
- [6] I. Daubechies, M. Defrise, and C. de Mol, “An iterative thresholding algorithm for linear inverse problems with a sparsity constraint,” *Communications on Pure and Applied Mathematics*, vol. 57, no. 11, pp. 1413–1457, 2004.
- [7] S. G. Mallat and Z. Zhang, “Matching pursuits with time-frequency dictionaries,” *IEEE Transactions on Signal Processing*, vol. 41, no. 12, pp. 3397–3415, 1993.
- [8] J. A. Tropp and A. C. Gilbert, “Signal recovery from random measurements via orthogonal matching pursuit,” *Institute of Electrical and Electronics Engineers. Transactions on Information Theory*, vol. 53, no. 12, pp. 4655–4666, 2007.
- [9] D. Needell and R. Vershynin, “Signal recovery from incomplete and inaccurate measurements via regularized orthogonal matching pursuit,” *IEEE Journal on Selected Topics in Signal Processing*, vol. 4, no. 2, pp. 310–316, 2010.
- [10] D. L. Donoho, Y. Tsaig, I. Drori, and J.-L. Starck, “Sparse solution of underdetermined systems of linear equations by stagewise orthogonal matching pursuit,” *Institute of Electrical and Electronics Engineers. Transactions on Information Theory*, vol. 58, no. 2, pp. 1094–1121, 2012.
- [11] W. Dai and O. Milenkovic, “Subspace pursuit for compressive sensing signal reconstruction,” *Institute of Electrical and Electronics Engineers. Transactions on Information Theory*, vol. 55, no. 5, pp. 2230–2249, 2009.
- [12] D. Needell and J. A. Tropp, “CoSaMP: iterative signal recovery from incomplete and inaccurate samples,” *Applied and Computational Harmonic Analysis. Time-Frequency and Time-Scale Analysis, Wavelets, Numerical Algorithms, and Applications*, vol. 26, no. 3, pp. 301–321, 2009.
- [13] J. Wang, S. Kwon, and B. Shim, “Generalized orthogonal matching pursuit,” *IEEE Transactions on Signal Processing*, vol. 60, no. 12, pp. 6202–6216, 2012.
- [14] J. Wang, S. Kwon, P. Li, and B. Shim, “Recovery of sparse signals via generalized orthogonal matching pursuit: a new analysis,” *IEEE Transactions on Signal Processing*, vol. 64, no. 4, pp. 1076–1089, 2016.
- [15] S. Satpathi, R. L. Das, and M. Chakraborty, “Improving the bound on the RIP constant in generalized orthogonal matching pursuit,” *IEEE Signal Processing Letters*, vol. 20, no. 11, pp. 1074–1077, 2013.
- [16] Y. Shen, B. Li, W. Pan, and J. Li, “Analysis of generalised orthogonal matching pursuit using restricted isometry constant,” *Electronics Letters*, vol. 50, no. 14, pp. 1020–1022, 2014.
- [17] B. Li, Y. Shen, Z. Wu, and J. Li, “Sufficient conditions for generalized orthogonal matching pursuit in noisy case,” *Signal Processing*, vol. 108, pp. 111–123, 2015.



Hindawi

Submit your manuscripts at
<https://www.hindawi.com>

