Journal of Information Hiding and Multimedia Signal Processing ©2018 ISSN 2073-4212
Ubiquitous International Volume **9**, Number **2**, March **2018**

A Modified Generalized Orthogonal Matching Pursuit Method for Compressed Sensing

Liquan Zhao, Yulong Liu
College of Information Engineering
Northeast Electric Power University
NO.169 Changchun Street, Chuanying District, Jilin City, China
zhao_liquan@163.com

Received May, 2017; revised October, 2017

Abstract. *An improved greedy algorithm was proposed in order to improve the reconstruction performance of the Generalized Orthogonal Matching Pursuit (GOMP) for sparse signals. In each iteration, the GOMP algorithm identifies multiple columns from the sensing matrix, in accordance with their matching degree with the residual . The indices of the columns are added to an estimated support set to reconstruct a sparse signal through solving a least-squares problem. The numbers of selected indices within each iteration are a fixed constant in the GOMP; however, the fixed constant is not suit to all iterations. Incorrect indices are selected when the fixed constant is larger than the number of the correct indices. Those selected incorrect indices thus reduce the reconstruct speed and accuracy, particularly when the fixed constant is much larger than the number of the correct indices. In order to mitigate weakness found within the GOMP, the proposed method firstly perform the primary election to the candidates with a specially chosen threshold. If the number of the candidates through the primary election is more than S, the proposed method will select S candidates from the candidate pool through the primary election and add their indices to the estimated support set. If the number of the candidates through the primary election is less than S, the proposed method will directly add the indices of all the candidates through the primary election to the estimated support set. The proposed method reduces the probability of selecting incorrect indices. The simulation results demonstrated the proposed algorithm has a better recovery performance than the original algorithm both in the probability of exact reconstruction and the time of, particularly in large signal problems.*
Keywords: Generalized Orthogonal Matching Pursuit; candidate; primary election; new improve method; threshold.

1. **Introduction.** The compressive sensing (CS)[1] theory has gained interests due to its ability to recover signals at a sampling frequency far lower than the Nyquist sampling rate. Different from the traditional Nyquist sampling theory, the CS includes three components: sparse representation, non-related linear measurement, and signal reconstruction. The reconstruction algorithm aims to recover the signals accurately from the measurements, thus being one of the most important components of the CS, as it determines the feasibility.

There has been many reconstruction algorithms proposed in order to reconstruct the sparse original sparse signal from the measurements reliably and accurately. The existing reconstruction algorithms can be divided into two major types : l_1-minimization and greedy pursuit algorithms. Common l_1-minimization approaches include basis pursuit (BP)[2], Gradient projection for sparse reconstruction (GPSR)[3], iterative thresholding

(IT)[4], and other algorithms. These algorithms demonstrate good performance within solving a convex minimization problem. These convex relaxation algorithms require minimal measurements, but they are more computationally complex.

An alternative approach is the greedy search approach, designed to further reduce the computational complexity of l_1-minimization approaches. The greedy algorithms identify the support of sparse vector in an iterative fashion, generating a series of locally optimal updates and construct an approximate signal based on the set of the chosen supports. Greedy algorithms have received an increase in attention due to their excellent performance and low cost in recovering sparse signals. The orthogonal matching pursuit algorithm (OMP) [5], which builds on the matching pursuit algorithm (MP)[6], is a well-known greedy algorithm with many applications. In OMP, the index of the column that is best correlated with the residual is chosen as a new element of the support within the iteration. The strategy of the OMP is to select one index for each iteration, which slows down the speed and hinders the reliability of the algorithm. In order to mitigate these OMP weaknessesvarious approaches have been proposed.

The regularized orthogonal matching pursuit algorithm (ROMP)[7] was developed to regularize the selected indices of the measurement matrix in order to improve the speed of OMP. The stagewise orthogonal matching pursuit (StOMP)[8] selects multiple indices in each iteration via a presupposed threshold. The subspace pursuit (SP)[9] and compressive sampling matching pursuit (CoSaMP)[10] proposed and boasts similar improvements. Both of these algorithms were proposed with the idea of backtracking; the differences bring that SP selects K indices for each iteration, while CoSaMP selects 2K.

The generalized orthogonal matching pursuit (GOMP), which selects $S(S \leq K)$ indices in each iteration, was proposed by Wang[11][12]. Compared to OMP, which selects only one index in each iteration, GOMP simply selects fixed multiple indices in each iteration to improve the empirical performance of the OMP as well as theoretical performance. The generalized OMP (GOMP) has received increasing attention in recent years with several papers being published on the analysis of the theoretical performance of GOMP [11-15]. However, GOMP has its weakness. It selects indices according a fixed number in each iteration, thus it may select many incorrect indices to the estimate support set. These incorrect indices reduce the recovery performance of algorithm. To mitigate these weaknesses, we propose a new method to modify GOMP by filtrating the candidates through a specially chosen threshold before they are selected. The simulation results suggest that the proposed algorithm has a better recovery performance than the original algorithm both in the probability of exact reconstruction and the time of reconstruction.

2. **Compressive Sensing.** The CS reconstructs the signal $x(x \in R^N)$ from compressed measurements $y = \Phi x \in R^M$ even when the system representation is underdetermined $(M < N)$. As a basic premise, compressive sensing requires that the signal x is a K-sparse signal. This means that if the signal is used as a dimensional vector $x(x \in R^N)$, there should be at most K no-zero elements in x. However, in practical applications, signals may not be sparse. When the target signal is not sparse, it has to be transformed into a sparse signal based on a set of sparse basis $\Psi = \{\phi_1, \phi_2, \phi_3......\phi_N\}$. In this case, can be defined as

$$x = \sum_{i=1}^{N} \alpha_i \phi_i = \Psi\alpha \tag{1}$$

where $\|\alpha\|_0 = K$. $\|.\|_0$ denotes the number of nonzero elements in a vector. Thus, the signal x is equivalently represented by K-sparse vector α under linear transformation Ψ in some domains.

Compressive sensing is regarded as a technique that automatically selects relevant information from signals by a measurement. In the theory, x is translated into a M-dimensional measurements y via a matrix multiplication with Φ. We describe it as

$$y = \Phi x \tag{2}$$

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

$$y = \Phi x = \Phi\Psi\alpha = A_{CS}\alpha \tag{3}$$

where $A_{CS} = \Phi\Psi \in R^{M\times N}$. Thus the problem to reconstruct x is transformed into recovering α from y measurements. As long as α is obtained, we can use (1) to obtain the original signal x. If Φ were a nonsingular square matrix, with $M = N$, we could easily recover x from y. Unfortunately, in most compressive sensing scenarios $M << N$. In above case, Eq(3) can be classified as an underdetermined system. The problem to reconstruct x from y is NP-hard. We can not to obtain an accurate reconstruction of x by solving the inverse transform of Φ. One way to solving this intractable computation can be described as an l-minimization problem:

$$\hat{\alpha} = \min ||\alpha||_1 \quad s.t. \quad y = A_{CS}\alpha. \tag{4}$$

An appropriate condition for exact recovery is that the matrix A_{CS} satisfies the condition of restricted isometry property (RIP) condition [1].

Definition 2.1. *Definition 1*

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

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

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

3. **GOMP Algorithm.** As a modification OMP algorithm, GOMP has similar principle with OMP. Both of them can be described as four steps: Identification, Augmentation, Estimation, and Residual update. The difference between of two algorithms lies in the identification step. In the identification step, OMP chooses a column of Φ that is maximally correlated with the residual in each iteration, where $\Phi \in R^{M\times N}$is the measurement. In the step, GOMP choose $S(S \ge 1)$ columns of Φ the largest correlated with the residual in each iteration. In the augmentation step, both the algorithms add the indices of that column to a list. In the estimation step, they use the indices in the list to obtain the estimation signal of x by solving the least square. Within the residual update step, they deduct the estimation signal of x this iteration from measurements, which generates a new residual used for the next iteration.

In the kth iteration, GOMP initially computes the correlation between the columns of the sensing matrix Φ and the residual vector r^{k-1} by $\Phi' r^{k-1}$, where r^{k-1} denote the residual vector in kth iteration and Φ' is a transpose matrix of Φ. Then indices of the columns corresponding to S maximal correlation are chosen as the new elements of the estimated support set Λ^k in each iteration, where Λ^k is the estimated support set in kth iteration. Then $\hat{x}^k$ is obtained using the least square method (LS), where $\hat{x}^k$ is the new approximation of x in kth iteration. The residual $r^k \in R^M$ is revised by subtracting $\Phi_{\Lambda^k}\hat{x}_{\Lambda^k}$ from y:

$$r^k = y - \Phi_{\Lambda^k}\hat{x}_{\Lambda^k}$$

where $y = \Phi x$. These operations are repeated until either the iteration number reaches the maximum $k_{\max} = \min(K, M/S)$ where K is the sparsity of x, or the l_2-norm of the residual falls below a threshold $\varepsilon(\left\|r^k\right\|_2 \leq \varepsilon$). To make the process clear, we describe the GOMP algorithm in Table according to[11][12]

It can be seen that that the GOMP only differs within the identification step of OMP, however the promotion to OMP within computational efficiency and recovery performance is great. This is due to the selection of multiple indices in GOMP, where multiple correct indices are added to the list and the algorithm is finished with much less iteration. We set the critical sparsity and the running time as two major evaluation criterions. The critical sparsity of a algorithm is the maximal sparsity level of the test sparse signals at which the perfect recovery is ensured. When exceed the point , the recovery rate of the algorithms drops below 100%. The simulation results reveal that the critical sparsity of the GOMP algorithm is larger than that of the ROMP, OMP, StOMP, and SP algorithms. Moreover, the running time of GOMP is less compare to above algorithms. Although GOMP is an outstanding algorithm, it has shortcomings. We will introduce the proposed method in next section.

We briefly summarize notations used in the table I. $\hat{x}$ is the final approximation of x. $\Phi_\Lambda \in R^{M\times|\Lambda|}$ is a submatrix of Φ that contains columns indexed by Λ. For example, if $\Phi = \{\phi_1, \phi_2, \phi_3, \phi_4\}$, and $\Lambda = \{1, 4\}$, then $\Phi_\Lambda = \{\phi_1, \phi_4\}$. $\Phi'$is a transpose matrix of Φ. If Φ is a full column rank, then $\Phi^\dagger = (\Phi'\Phi)^{-1}\Phi'$ is the pseudoinverse of Φ. $\hat{x}_\Lambda \in R^{|\Lambda|}$ is the estimated vector of x, which elements based upon the indices of Λ. At the kth iteration, we use Λ^k, $\hat{x}^k$ and r^k denote the estimated support, the estimated sparse signal, and the residual vector, respectively.

TABLE 1. Gomp Algorithm

Input:	measurements $y \in R^M$, sensing $\Phi \in R^{M\times N}$, sparsity K, number of indices of columns for each selection $S(S \leq K)$.
Initialize:	iteration count $k = 0$, residual vector $r^0 = y$, estimated support set $\Lambda^0 = \emptyset$.
While	$\|\|\mathrm{r}^k\|\| > \varepsilon$ 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 $\{\varphi(i)\}_{i=1,2,3......S}$ corresponding to the entries. **(Augmentation)** $\Lambda^k = \Lambda^{k-1} \cup \{\varphi(1), \varphi(2)......\varphi(S)\}$. **(Estimation of x_{Λ^k})** $\hat{x}_{\Lambda^k}=\Phi^\dagger_{\Lambda^k} y$. **(Residual Update)** $r^k = y - \Phi_{\Lambda^k}\hat{x}_{\Lambda^k}$.
End	
Output	The estimated signal $\hat{x}_k = \hat{x}_{\Lambda^k}$, satisfying $\hat{x}_{\{1,...,n\}-\Lambda^k} = 0$.

4. **Proposed Algorithm.** For the greedy algorithms, it is important to generate an estimate of the correct support set. We assume the correct support set is T. Let $\Omega = \{1, 2, ..., \mathrm{n}\}$ be the column indices of matrix Φ , then $T = \{i | i \in \Omega, x_i \neq 0\}$ denote the support of vector x . Λ_k is the estimate of T in kth iteration. The goal is to obtain

$$\Lambda^k = \underset{\mathrm{T}:|\mathrm{T}|=K}{\arg\min} \|\hat{x}_{\Lambda^k} - \hat{x}_{\mathrm{T}}\|_2$$

where $\hat{x}_\Lambda \in R^{|\Lambda|}$ is the estimated vector of x which elements based upon the indices of Λ and $|\Lambda|$ is the cardinality of Λ.

GOMP selects multiple indices for each iteration, with the expectation that multiple correct indices are added to Λ^k in order to finish the reconstruction with a smaller number of iterations.

The idea of GOMP is imperfect. We suppose there are I correct indices should be selected in the kth iteration. According to strategy of GOMP, S indices would be chosen and added to Λ^k. Then $(S-I)$ incorrect indices are added to Λ^k. Those incorrect indices of Λ^k, once be selected, will remain in the list throughout the remainder of the reconstruction process. Those incorrect indices will reduce the computational efficiency and recovery performance. This is aggravated when the GOMP selects a S much larger than I, and the weakness of the GOMP becomes more apparent.

In order to mitigate the algorithmic weakness of GOMP, we propose a modifications of GOMP with the thresholding scheme initially proposed in StOMP[8] and describe the proposed method in Table II. StOMP regards the noiseless underdetermined problems as a noisy well-determined problem. In each iteration, StOMP regards the correct indices and incorrect indices as true signal and "noise". StOMP then sets an appropriate threshold to identify the true signal from the mixture. We describe the process in the kth iteration as

$$u_k = \Phi' r^{k-1},$$

$$J_k = \{j : |u_k(j)| > t_k \sigma_k\}.$$

where σ_k is a formal noise level and t_k is a threshold parameter. The formal noise level $\sigma_k = \|r_k\|_2 \big/ \sqrt{M}$, and typically the threshold parameter take values in the range $2 \leq t_k \leq 3$. The thresholds are specifically chosen based on the assumption of Gaussianity. Gaussianity assumes if $\Phi \in R^{M\times N}$ is a matrix from Uniform Spherical ensemble (USE), the element in u_k has approximately a Gaussian distribution with mean 0. When M and N are both large, this situation will be more accurate, meaning that the method will be more effective on large-scale signal situation.

In the kth iteration, the proposed method firstly computes the correlation between the columns of the sensing matrix Φ and the residual vector r^{k-1},

$$u_k = \Phi' r^{k-1}.$$

Then to yield a set J_k with a hard thresholding

$$J_k = \{j : |u_k(j)| > t_k \sigma_k\}.$$

If $|J_k| \geq S$, the proposed method yield the set $U_k = \{|u_k(i)|\, i \in \{1, 2...\mathrm{N}\}\}$ and $V_k = \{|u_k(i)|\,, i \in J_k\}$. It is obvious that V_k is a subset of U_k. Then the proposed method select S largest elements of V_k and find their indices in U_k. The indices are added to the set Q_k. We merge the newly selected Q_k with the previous support estimate, thereby updating the estimate support set:

$$\Lambda^k = \Lambda^{k-1} \cup Q_k.$$

If $|J_k| < S$, we choose J_k as the subset of newly selected, thereby updating the estimate:

$$\Lambda^k = \Lambda^{k-1} \cup J_k$$

Further then to obtain new approximation of , using the least square method (LS):

$$\hat{x}_{\Lambda^k} = \Phi^\dagger_{\Lambda^k} y$$

where $\Phi^{\dagger} = (\Phi'\Phi)^{-1}\Phi'$ is the pseudoinverse of Φ and Φ' is a transpose matrix of Φ. Finally, the residual r^k is revised by subtracting $\Phi_{\Lambda^k}\hat{x}_{\Lambda^k}$ from y:

$$r^k = y - \Phi_{\Lambda^k}\hat{x}_{\Lambda^k}$$

Generally speaking, the proposed method adds a threshold as supplementary conditions in the identification step. When the that the GOMP selected is oversize in some iterations, the proposed method obtains a more reliable estimate set by filtrating the candidates to replace the estimate set yields by the identification strategy of the GOMP. Once the incorrect indices are selected into the estimate support set, they will not be removed in subsequence step. These incorrect indices will reduce both accuracy and speed of algorithm. The proposed method can effectively avoid incorrect indices are added to the estimate support set, thus improving the reconstruction performance of the GOMP.

Table 2. Improved Method of GOMP

Input:	measurements $y \in R^M$,
	sensing $\Phi \in R^{M\times N}$,
	sparsity K,
	number of indices of columns for each selection $S(S \le K)$.
Initialize:	iteration count $k = 0$,
	residual vector $r^0 = y$,
	estimated support set $\Lambda^0 = \emptyset$.
	newly subset J_k=$\emptyset$, $Q_k = \emptyset$.
For	$k = 1 : K$
	1. calculate $u_k = \Phi' r^{k-1}$
	2. yield a newly set $J_k = \{j : \lvert u_k(j)\rvert > t_k\sigma_k\}$.**(Identification)**
	3. If $\lvert J_k\rvert \ge S$, yield the $U_k = \{\lvert u_k(i)\rvert\, i \in \{1, 2...\text{N}\}\}$ and the set $V_k = \{\lvert u_k(i)\rvert, i \in J_k\}$. Select S largest elements of V_k and find their indices in U_k. Add these indices to a set Q_k, $\Lambda^k = \Lambda^{k-1} \cup Q_k$; If $\lvert J_k\rvert < S$, $\Lambda^k = \Lambda^{k-1} \cup J_k$.**(Identification and Augmentation)**
	4. $\hat{x}_{\Lambda^k}$=$\Phi^{\dagger}_{\Lambda^k} y$.**(Estimation of x_{Λ^k})**
	5. $r^k = y - \Phi_{\Lambda^k}\hat{x}_{\Lambda^k}$.**(Residual Update)**
	6. If $\lVert \text{r}^{\text{k}}\rVert < \varepsilon$, quit the iteration.
End	
Output	The estimated signal $\hat{x}_k = \hat{x}_{\Lambda^k}$, satisfying $\hat{x}_{\{1,\dots,n\}-\Lambda^k} = 0$.

Annotation: The formal noise level $\sigma_k = \|r_k\|_2 \big/ \sqrt{M}$, where the threshold typically has values in the range $2 \le t_k \le 3$.

5. **Simulation and discussion.** The following will demonstrate the reconstruction performance of the proposed algorithm with sparse signals. In each trial, we generated a K-sparse vector $x \in R^N$ whose support is chosen at random. Additionally, we constructed a sensing matrix $\Phi \in R^{M\times N}$ with entries drawn independently from a Gaussian distribution $N(0, 1/M)$, with $t_k = 2.5$ empirically chosen. In order to compare the reconstruction performance with the signal with different scales, our trials adapt two types of x and Φ($M = 128$, $N = 256$ and $M = 128$, $N = 512$). We used MATLAB 7.0 with a quad-core 64-bit processor in a Window 7 environment, with each algorithm repeating 400 times and the probability of the exact reconstructions and the average running time in each K was recorded.

In our simulation, we set the critical sparsity and the running as two major criterions. The critical sparsity of a algorithm is the maximal sparsity level of the test sparse signals at which the perfect recovery is ensured. When exceed the point, the recovery rate of the algorithms drops below 100%. Clearly, higher critical sparsity and less running time imply better empirical reconstruction performance.

5.1. **Experiment 1.** In the experiment, we provide recovery performance as a function of the sparsity level K and set $M = 128$, $N = 512$. We compared our proposed method ($S = 3$), GOMP ($S = 3$), and others prime greedy algorithms. In Figure 1, the simulation results reveal that the critical sparsity of the proposed method and the GOMP algorithm was larger than that of the ROMP, OMP, StOMP and SP algorithms, and the proposed method almost had no difference within the GOMP. Fig. 2 provides the results in running time for recovery algorithms. It is seen that the running time of our proposed method, GOMP, StOMP, ROMP, SP, and OMP is more or less similar when $K \leq 35$. When the signal vector becomes less sparse ($K > 35$), the difference is gradually apparent. The running time of the SP increases fastest in all algorithms. Then the running time of the GOMP also has a faster increase. The propose method, OMP, StOMP, and ROMP keep a steady tendency with similar running time. Observe carefully, the running time of the proposed method is smaller than OMP and larger than ROMP and StOMP, slightly. Overall, the proposed method perform is competitive both in critical sparsity and running time.

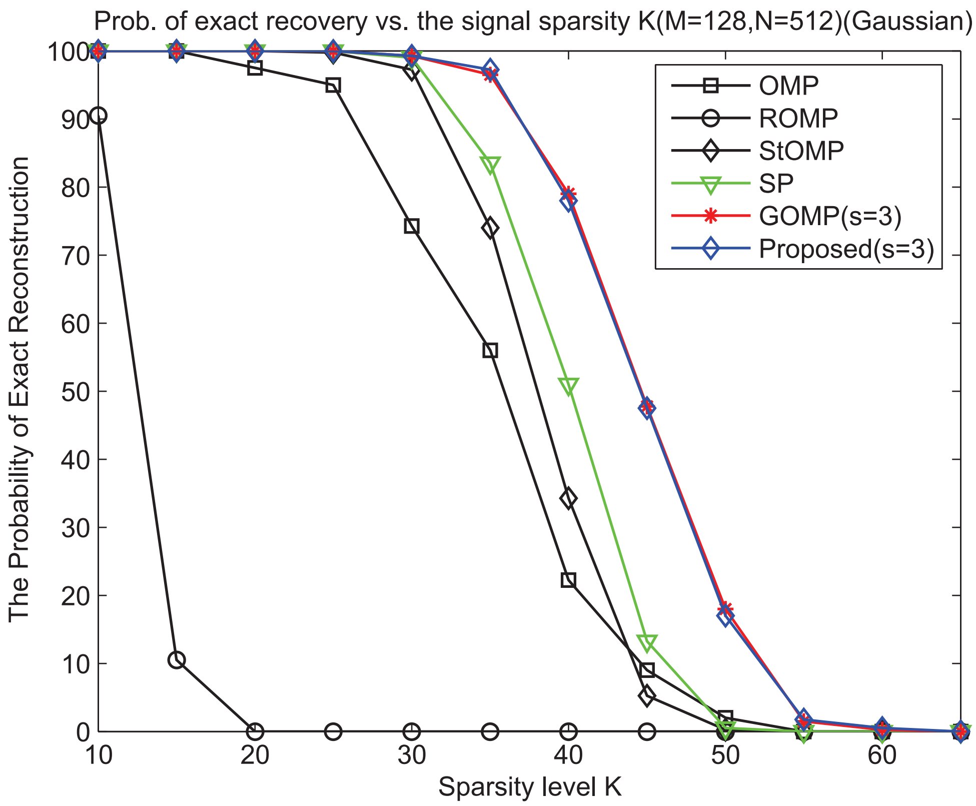

FIGURE 1. Reconstruction performance as a function of sparsity K.

5.2. **Experiment 2.** In this experiment, we compared our proposed method and the GOMP. We set two signals with different sizes ($N = 256, 512$) and three different $S(S =$

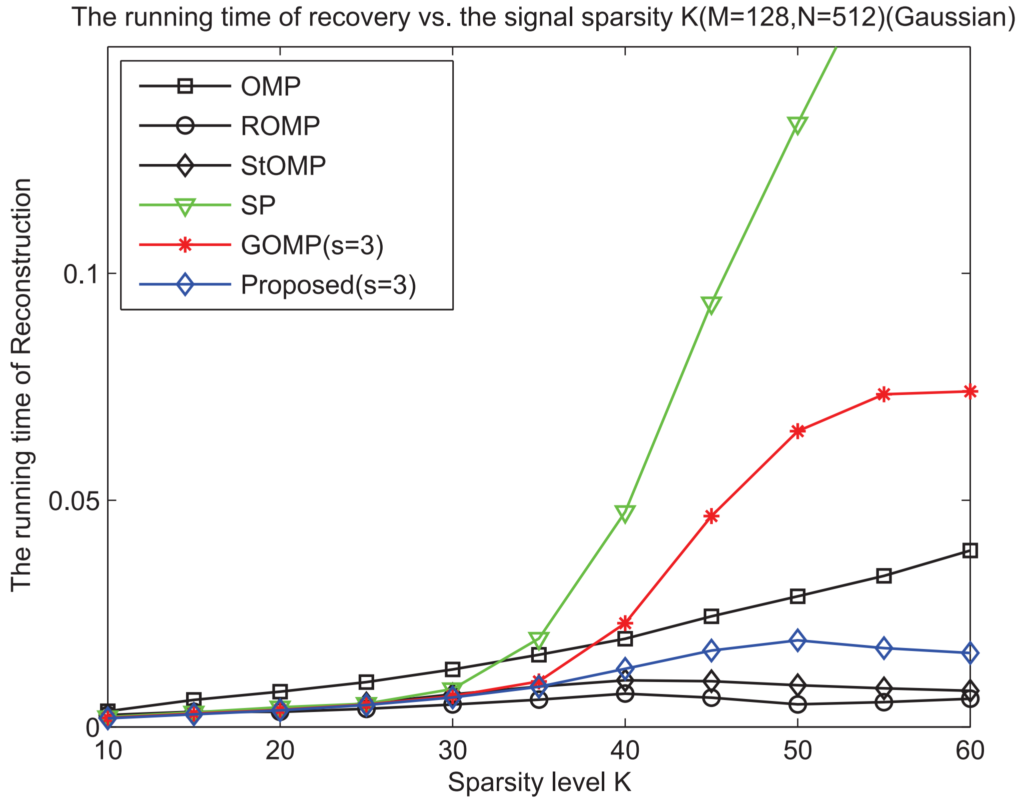

FIGURE 2. Running time as a function of sparsity K.

3, 10, 20). In order to compare two algorithms with different S, we set the mean of two algorithms to be observed in the experiment. Fig.3 and Fig.4 provide results where

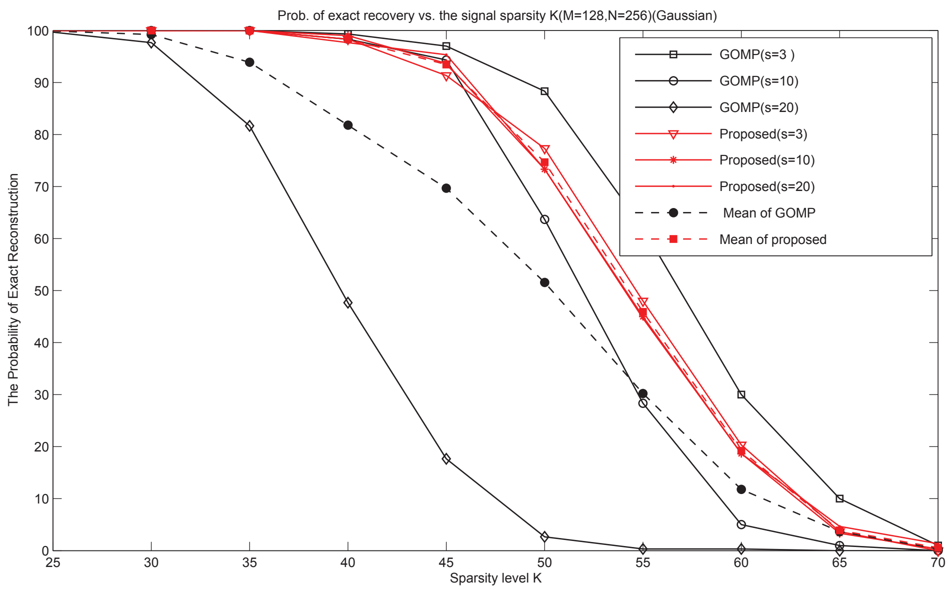

FIGURE 3. Reconstruction performance of GOMP and the proposed method(M=128,N=256)

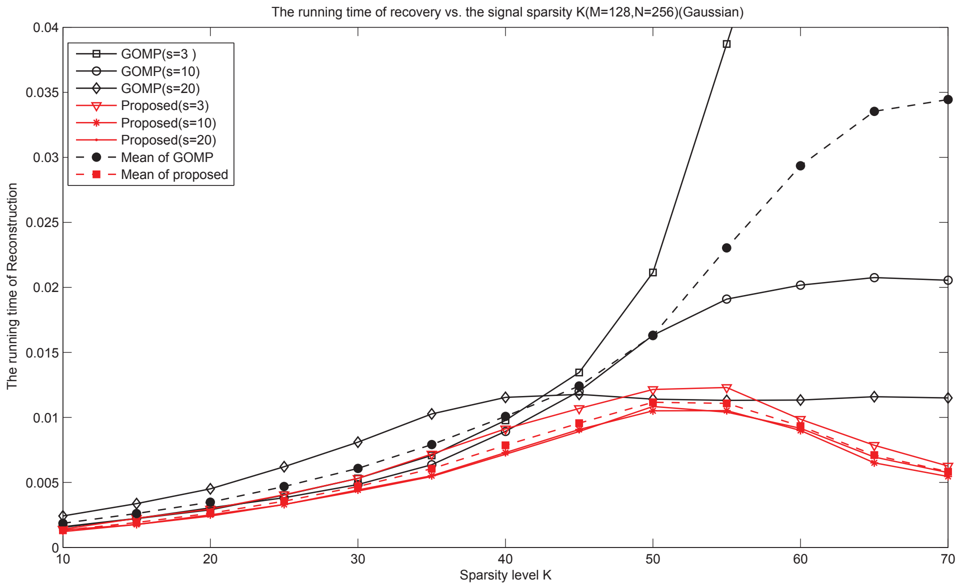

FIGURE 4. The average running time of GOMP and the proposed method(M=128,N=256)

$M = 128$ and $N = 256$. From Fig.3 we can observe that with the smaller S, the critical sparsity in GOMP is higher. However the proposed method with different S have nearly analogous performance. The critical sparsity of the proposed methods are between GOMP ($S = 3$) and GOMP ($S = 10$) and farther larger than GOMP ($S = 20$). The mean critical sparsity of the proposed method is higher than GOMP. Fig.4 provides the results of the running time. The results of two methods are more or less similar when $K \leq 45$. When $K > 45$ the running time of GOMP ($S = 3$) increases quickly, and the running time of GOMP($S = 10$) slows. The GOMP ($S = 20$) has similar performance to the proposed method ($S = 3$). Among algorithms being tested, the running time of the proposed methods ($S = 10, 20$) is the shortest. The mean running time of the proposed method is also smaller than GOMP.

Fig.5 and Fig.6 provide results when $M = 128$ and $N = 512$. From Fig.5 we can observe that both the proposed method and GOMP show the trend that the smaller S , the higher critical sparsity. The GOMP ($S = 3$) and the proposed method ($S = 3$) perform almost alike and have the largest critical sparsity among algorithms under test. The proposed method ($S = 20$) has similar performance to the GOMP ($S = 10$), where both exhibit better than the GOMP ($S = 20$) and slightly worse to the proposed method ($S = 10$). The mean critical sparsity of the proposed method is also higher than GOMP. From Fig.6 We can see that the main difference in running time is visible when $K > 35$. In this case, the running time of GOMP($S = 3$) increases fast and the GOMP($S = 20$) perform best in running time. But we note that when $K = 35$, the probability of exact reconstruction only 10% while the probability of others over 90%. So, to discuss the performance of GOMP ($S = 20$) when $K > 35$ is meaningless. With the exception of the GOMP ($S = 20$), the proposed method ($S = 3$) has smallest running time among the algorithms. The running time of the proposed method ($S = 10, 20$) slightly bigger than GOMP ($S = 10$).The mean running time of the proposed method is also smaller than GOMP.

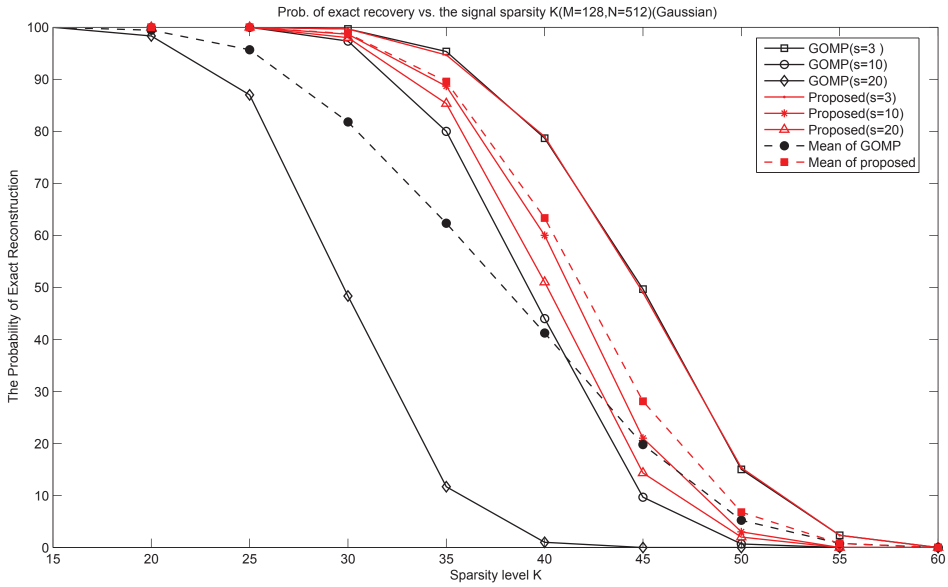

FIGURE 5. Reconstruction performance of GOMP and the proposed method(M=128,N=512)

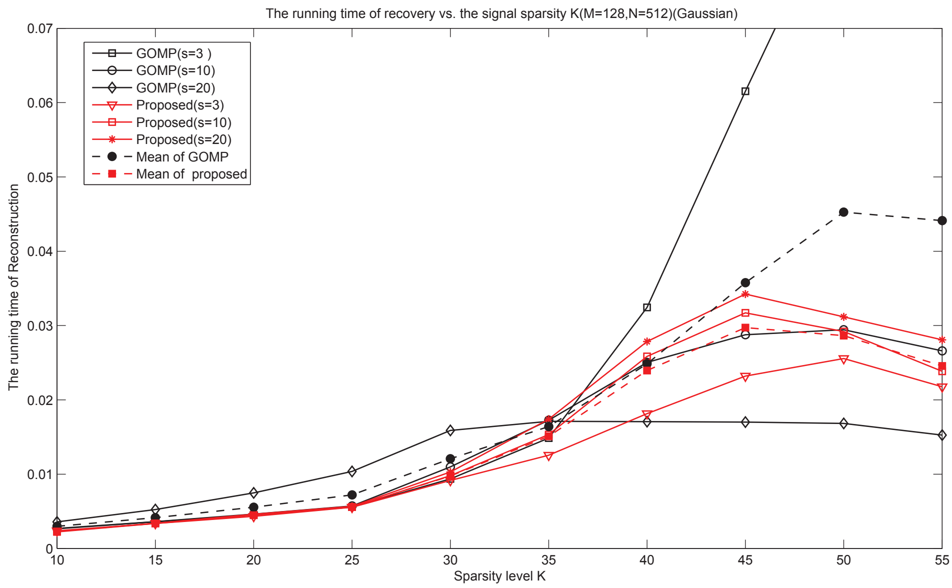

FIGURE 6. The average running time of GOMP and the proposed method(M=128,N=512)

Simulation results show that the proposed improved method performs well, and that it outperforms the GOMP in terms of both critical sparsity and running time and the proposed improved is more effective on large-scale signal situation.

6. **Conclusion.** In this paper, a novel method for sparse signal reconstruction is proposed. The method adds a threshold as supplementary conditions in the identification step. When the number of selected indices experiences an increase in iteration, our method

obtains a more reliable estimate set by filtrating the candidates. The method can avoid selecting incorrect indices into the estimated support set. Our proposed reconstruction algorithm not only performs well with 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 prove that the proposed method had superior performance than that of GOMP both with regard to critical sparsity and running time, particularly in large signal problems.

REFERENCES

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

[2] 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.

[3] 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 of Selected Topics in Signal Processing*, vol.1, no.4, pp.586–597, 2007.

[4] I. Daubechies, M. Defrise, and C. D. Mol, An iterative thresholding algorithm for linear inverse problems with a sparsity constraint, *Communications on Pure and Applied Mathematics*, vol.57, no.11, pp.1413C-1457, 2004.

[5] 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.

[6] S. Mallat and, Z. Zhang, Matching pursuit with time-frequency dictionaries, *IEEE Transactions on Signal Processing*, vol.41, no.12, pp.3397–3415, 1993.

[7] D. Needell and, R. Vershynin , Signal recovery from incomplete and inaccurate measurements via Regularized Orthogonal Matching Pursuit, *IEEE Journal of Selected Topics in Signal Processing*, vol.4, no.2, pp.310–316, 2007.

[8] D. L. Donoho, Y. Tsaig, and I. Drori, Sparse Solution of Underdetermined Systems of Linear Equations by Stagewise Orthogonal Matching Pursuit, *IEEE Transactions on Information Theory*, vol.58, no.2, pp.1094–1121, 2012.

[9] W. Dai, O. R. Milenkovic, and A. Spanias, Subspace pursuit for compressive sensing signal reconstruction, *IEEE Transactions on Information Theory*, vol.55, no.5, pp.2230–2249, 2009.

[10] 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, 2008.

[11] J. Wang, S. Kwon, and B. Shim, Generalized orthogonal matching pursuit, *IEEE Transaction on Signal Processing* , vol.60, no.12, pp.6202–6216, 2012.

[12] J. Wang, S. Kwon, and P. Li, 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.

[13] 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.20, pp.1074–1077, 2013.

[14] Y. Shen, B. Li, and Pan. W, et al, Analysis of generalised orthogonal matching pursuit using restricted isometry constant, *Electronics Letters*, vol.54, no.14, pp.1020–1022, 2014.

[15] Y. Shen, B. Li, and Z. Wu, Sufficient conditions for generalized Orthogonal Matching Pursuit in noisy case, *Signal Processing*, vol.108, no.108, pp.111–123, 2015.