

Research Article

A Hybrid Orthogonal Forward-Backward Pursuit Algorithm for Partial Fourier Multiple Measurement Vectors Problem

Haiqiang Liu ¹, Gang Hua ¹, Aichun Zhu ², Hongsheng Yin ¹, and Yonggang Xu ¹

¹China University of Mining and Technology, Xuzhou 221116, China

²Nanjing Tech University, Nanjing 221800, China

Correspondence should be addressed to Gang Hua; ghua@cumt.edu.cn

Received 20 November 2017; Revised 9 May 2018; Accepted 3 June 2018; Published 17 July 2018

Academic Editor: J.-C. Cortés

Copyright © 2018 Haiqiang Liu et al. 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.

In solving the partial Fourier Multiple Measurement Vectors (FMMV) problem, existing greedy pursuit algorithms such as Simultaneous Orthogonal Matching Pursuit (SOMP), Simultaneous Subspace Pursuit (SSP), Hybrid Matching Pursuit (HMP), and Forward-Backward Pursuit (FBP) suffer from low recovery ability or need sparsity as a prior information. This paper combines SOMP and FBP to propose a Hybrid Orthogonal Forward-Backward Pursuit (HOFBP) algorithm. As an iterative algorithm, each iteration of HOFBP consists of two stages. In the first stage, α indices selected by SOMP are added to the support set. In the second stage, the support set is shrunk by removing β indices. The choice of α and β is critical to the performance of this algorithm. The simulation results showed that, by using proper parameters, HOFBP has better performance than other greedy pursuit algorithms at the expense of more computing time in some cases. HOFBP does not need sparsity as a prior knowledge.

1. Introduction

Compressed Sensing (CS) [1–3] is a new information collection theory which has broken through the Nyquist sampling theorem. The Multiple Measurement Vector (MMV) problem [4] is a natural extension of CS to deal with multiple signals. It has potentials in many practical problems, such as underwater acoustic channel estimation [5], source localization [6], Direction-Of-Arrival (DOA) [7] as well as image super-resolution [8], face recognition [9], and image reconstruction [10]. In MMV problems, if the partial Fourier measurement matrix is used to measure signals, we have a partial Fourier MMV problem.

Many algorithms were proposed to solve the MMV problem, such as Multiple Signal Classification algorithms [10–13], Bayesian learning algorithms [14–16], and greedy pursuit (GP) algorithms [17–20] as well as l_1 -SVD [6] and q -thresholding algorithm [21]. Generally, GP algorithms are easily implemented and have low computational complexity. It is known that the GP algorithms are based on the iterative selection of the strongest components in the measurements projected onto the measurement matrix columns. Different

from Gaussian random measurement matrix, partial Fourier matrix has the property that its adjoining atoms (columns of the measurement matrix) are strongly correlated [17, 22]. Because of this, greedy pursuit (GP) algorithms have some problems in solving the partial Fourier MMV problem. Simultaneous Orthogonal Matching Pursuit (SOMP) [18] lacks a mechanism to reevaluate the selected indices. Once a wrong index is selected, there is no chance to remove it. Simultaneous Subspace Pursuit (SSP) [19] and Simultaneous Compressive Sampling Matching Pursuit (SCoSaMP) [20] select multiple atoms in each selection. This makes them weak in distinguishing adjoining atoms. G. Li and R. J. Burkholder [17] combine SOMP and SSP to propose the Hybrid Matching Pursuit (HMP) algorithm. However, in our simulations, HMP's performance is not always good enough. Moreover, SSP, SCoSaMP, and HMP need sparsity as a prior information, which is impossible to know a prior for most practical applications. Forward-Backward Pursuit (FBP) [23] does not need the sparsity constraint as a prior information. However, it is also weak in distinguishing adjoining atoms. As stated in [24], the Bayesian-based CS reconstruction algorithms run in the opposite direction to GP algorithms.

In the Bayesian reconstruction, the sparsity is reduced from its maximal value by using hyperparameters. In this case, if a component is missed, then the signal cannot be recovered. While the inclusion of a nonexistent component will only influence the noise in the resulting reconstruction, the omission of such a component will prevent correct reconstruction.

In this paper, we combine SOMP and FBP to propose a Hybrid Orthogonal Forward-Backward Pursuit (HOFBP) algorithm. Inheriting the advantages of both SOMP and FBP, HOFBP has high-resolution in distinguishing adjoining atoms, as well as a mechanism to reevaluate the selected indices, and does not require sparsity as a prior information. Simulation results show that, by using proper parameters, HOFBP has better performance than other GP algorithms at the expense of more computing time in some cases.

The rest of this paper is organized as follows. In Section 2, we introduce some notation and concepts. In Section 3, we introduce related GP algorithms, including SOMP, SSP, HMP, and FBP. In Section 4, we propose the HOFBP algorithm. In Section 5, the computational complexity of HOFBP is analyzed. In Section 6, we provide the simulation results. Conclusions are stated in Section 7.

2. Notation and Concepts

In this paper, we use the following notations. L denotes the number of signals. $S \triangleq \{1, 2, \dots, N\}$ denotes the set of indices which correspond to the rows of the joint signal \mathbf{X} . For the matrix $\mathbf{A} \in \mathbf{R}^{M \times N}$, \mathbf{A}_j represents the j -th column of \mathbf{A} , and $\mathbf{A}_{(i)}$ represents the i -th row of \mathbf{A} . For the set $G = \{1, 2, \dots, q\}$, $|G|$ denotes the cardinality of G , \mathbf{A}_G represents the matrix composed of the columns $\{\mathbf{A}_j\}_{j \in G}$, and $\mathbf{X}_{(G)}$ represents the matrix composed of the rows $\{\mathbf{A}_{(i)}\}_{i \in G}$. $\Lambda(\mathbf{X}) \triangleq \{1 \leq i \leq N \mid \mathbf{X}_{(i)} \neq \mathbf{0}\}$ denotes the support set, which records the indices (positions of nonzero rows in \mathbf{X}). If $|\Lambda(\mathbf{X})| \leq K \ll M$, \mathbf{X} is called jointly sparse, and K is called the sparsity. In this paper, we assume $|\Lambda(\mathbf{X})| = K$. \mathbf{A}^H represents the conjugate transpose matrix of \mathbf{A} . $\mathbf{A}^\dagger = (\mathbf{A}^H \mathbf{A})^{-1} \mathbf{A}^H$ represents the pseudoinverse matrix of \mathbf{A} . $\|\mathbf{A}\|_F$ denotes the Frobenius norm of \mathbf{A} . $\text{span}(\mathbf{A}_G)$ denotes the space spanned by the columns of \mathbf{A}_G .

2.1. Single Measurement Vector (SMV). The measurement process of CS can be denoted as $\mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{n}$, where $\mathbf{x} = (x_1, x_2, \dots, x_N)^T \in \mathbf{R}^N$ is a sparse signal, $\mathbf{A} \in \mathbf{R}^{M \times N}$ ($M < N$) is the measurement matrix, $\mathbf{y} = (y_1, y_2, \dots, y_M)^T \in \mathbf{R}^M$ is the measurement vector, and $\mathbf{n} = (n_1, n_2, \dots, n_M)^T \in \mathbf{R}^M$ is the additive noise. Given \mathbf{A} and \mathbf{y} , estimating \mathbf{x} from \mathbf{y} is named as the Single Measurement Vector (SMV) problem [1–3, 25]. l_1 minimization algorithms [26–28] and GP algorithms are proposed to solve the SMV problem. There are many GP algorithms, such as Orthogonal Matching Pursuit (OMP) [29], Subspace Pursuit (SP) [30], Compressive Sampling Matching Pursuit (CoSaMP) [31], Forward-Backward Pursuit (FBP) [23] as well as Regularized Orthogonal Matching Pursuit (ROMP) [32], Stagewise Orthogonal Matching Pursuit (StOMP) [33], and Generalized Orthogonal Matching Pursuit (gOMP) [34].

2.2. Multiple Measurement Vector (MMV). The MMV problem is an extension of the SMV problem. In the MMV problem, the measurement matrix \mathbf{A} measures L signals, which can be denoted as

$$\mathbf{y}_l = \mathbf{A}\mathbf{x}_l + \mathbf{n}_l, \quad l = 1, 2, \dots, L \quad (1)$$

where $\mathbf{x}_l \in \mathbf{R}^N$ represents the l -th signal, $\mathbf{y}_l \in \mathbf{R}^M$ denotes the l -th measurement vector, $\mathbf{A} \in \mathbf{R}^{M \times N}$ ($M < N$) denotes the measurement matrix, and \mathbf{n}_l denotes the additive noise. The measurement process can be rewritten as $\mathbf{Y} = \mathbf{A}\mathbf{X} + \mathbf{Noise}$, where $\mathbf{Y} = [\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_L]$, $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_L]$, and $\mathbf{Noise} = [\mathbf{n}_1, \mathbf{n}_2, \dots, \mathbf{n}_L]$. If only K rows of \mathbf{X} are nonzero vectors, \mathbf{X} is called jointly sparse and the sparsity is K . Supposing \mathbf{X} is jointly sparse, the problem of recovering \mathbf{X} from \mathbf{Y} , which is also called the MMV problem, can be formulated as

$$\begin{aligned} \min_{\mathbf{X}} \quad & |\Lambda(\mathbf{X})| \\ \text{s.t.} \quad & \|\mathbf{Y} - \mathbf{A}\mathbf{X}\|_F < \varepsilon, \end{aligned} \quad (2)$$

where $\varepsilon \geq 0$ is the error bound.

2.3. Partial Fourier Measurement Matrix. In MMV, many measurement matrices can be used to measure signals, such as Gaussian random matrix [1], partial Fourier matrix [2, 3], and Bernoulli random matrix [2] as well as deterministic measurement matrices [35, 36].

If Gaussian random matrix is used to measure signals, we call the MMV problem as Gaussian random MMV problem. If the partial Fourier measurement matrix is used to measure signals, we call the MMV problem as partial Fourier MMV problem.

The elements of the Gaussian random measurement matrix follow a Gaussian distribution with mean $\mu = 0$ and variance $\sigma^2 = 1/M$. The Gaussian random measurement matrix can be defined as

$$\mathbf{A}_{p,s} \sim N\left(0, \frac{1}{M}\right), \quad (3)$$

$$p \in \{0, 1, \dots, M-1\}, \quad s \in \{0, 1, \dots, N-1\}$$

The Gaussian random measurement matrix is a random projection matrix [37–39], which is widely used in machine learning and dimensionality reduction.

The discrete Fourier transform (DFT) matrix is defined as

$$\Psi_{w,z} = \frac{1}{\sqrt{N}} \exp\left(-i \frac{2\pi w z}{N}\right), \quad (4)$$

$$w, z \in \{0, 1, \dots, N-1\}$$

The partial Fourier measurement matrix is formed by randomly choosing M rows from Ψ , where $M < N$.

The elements of Gaussian random measurement matrix follow Gaussian distribution. Its adjoining atoms are weakly

correlated. The measurement matrix used for radar imaging is typically a Fourier-like measurement matrix. As illustrated in [17, 22], compared with Gaussian random measurement matrix, the adjoin atoms of the Fourier-like measurement matrix are strongly correlated.

As GP algorithms iteratively select the atoms with the largest correlation to the measurement signal, it is harder for them to distinguish adjoining atoms of partial Fourier measurement matrix than that of Gaussian random measurement matrix. Our algorithm solves this problem by combining SOMP and FBP.

What should be pointed out is that because Gaussian random measurement matrix has not the problem as partial Fourier measurement matrix, our algorithm has no advantage over some GP algorithms in solving the Gaussian random MMV problem. The experimental results in the Appendix illustrate this aspect.

3. Greedy Pursuit Algorithms

In all greedy pursuit algorithms, a two-step strategy is adopted to solve problem (2). Firstly, the support set Λ is estimated. Then, the nonzero elements of the signal are estimated by using the least squares method, and the solution of problem (2) can be calculated as

$$\begin{aligned}\widehat{\mathbf{x}}_{l(\Lambda)} &= \mathbf{A}_\Lambda^\dagger \mathbf{y}_l \\ \text{and } \widehat{\mathbf{x}}_{l(S-\Lambda)} &= \mathbf{0},\end{aligned}\quad (5)$$

where $l = 1, 2, \dots, L$ and $S \triangleq \{1, 2, \dots, N\}$.

To simplify the description of the following algorithms, we define two helping functions. The function `max_indices()` selects a certain number of indices which correspond to the largest amplitude entries of a vector. The function `min_indices()` selects a certain number of indices which correspond to the smallest amplitude entries of a vector.

$$\text{max_indices}(\mathbf{a}, \alpha) \triangleq \{\alpha \text{ indices which correspond to the largest amplitude entries of the vector } \mathbf{a}\}. \quad (6)$$

$$\text{min_indices}(\mathbf{a}, \beta) \triangleq \{\beta \text{ indices which correspond to the smallest amplitude entries of the vector } \mathbf{a}\}. \quad (7)$$

SOMP is summarized in Algorithm 1. In SOMP, the support set expands iteratively. In each iteration, the index corresponding to the atom which is most related to the residual vectors is added to the support set. If the sum of all the residual vectors' l_2 -norms is larger than the termination threshold, we will terminate the iteration process, output the support set Λ , and then estimate the signals by using the least squares method as defined in (5); otherwise, the iteration process continues.

SSP is summarized in Algorithm 2; it searches the support set iteratively. Each iteration is divided into index selection stage and backtracking stage. In the index selection stage, K indices corresponding to the atoms which are most related to all the residual vectors are added to the support set. In this way, the size of the support set expands to $2K$ (except on the first iteration). In the backtracking stage, L measurement vectors are individually projected onto the subspace spanned by the $2K$ atoms. Then, the support set is refined by selecting K indices which correspond to the largest entries in the fusion result of L projection coefficients. Once the sum of all the residual vectors' l_2 -norms does not decrease, the iteration process is terminated and the signals can be estimated by (5).

Similar to SSP, FBP is also a two-stage algorithm. It has two parameters: forward step size α and backward step size β , where $\beta < \alpha$. In the first stage, α indices are added to the support set. In the second stage, β indices corresponding to the atoms which have the smallest contributions to projection coefficients are removed from the support set. Because $\beta < \alpha$, the support set expands with the step size $\alpha - \beta$. When the residual energy is less than a threshold, the iteration process is terminated. The choices of α and β are critical to the performance of FBP. According to the simulation results in [23], $\alpha \in [0.2M, 0.3M]$ and $\alpha - \beta = 1$ lead to the best

performance. In case of a failure, in order to avoid too many iterations, the size of the estimated support set is controlled by the parameter T , which is usually set as M .

HMP combines the strengths of SOMP and SSP. It also searches the support set iteratively. Each iteration of HMP can be divided into index selection stage and backtracking stage. In the index selection stage, it utilizes the strategy of SOMP to select indices. Firstly, L solutions are obtained by individually applying OMP on L residual vectors. Then, K indices corresponding to the largest amplitude entries of L solutions' fusion result are added to the support set. In the backtracking stage, HMP utilizes the reevaluation mechanism of SSP to reevaluate the selected indices. Firstly, L projection coefficients are obtained by individually projecting measurement vectors onto the subspace spanned by $\mathbf{A}_{\Lambda_{temp}}$, where Λ_{temp} is the support set. Then, the support set is refined by selecting K indices which correspond to the largest entries in the fusion result of the L projection coefficients. Once the sum of all the residual vectors' l_2 -norms no longer decreases, the iteration process is terminated and the signals can be estimated by (5); otherwise, the iteration process continues.

4. HOFBP

4.1. Strengths and Drawbacks of SOMP, SSP, FBP, and HMP. From the description of SOMP and SSP, we can see that SOMP and SSP have both strengths and drawbacks in solving the partial Fourier MMV problem. SOMP selects one index in each iteration, which ensures the strong ability in distinguishing adjoining atoms. However, it lacks a mechanism to reevaluate the selected indices. Once a wrong index is selected, it will lie in the support set forever.

Input: the measurement vectors \mathbf{y}_l where $l = 1, 2, \dots, L$, the measurement matrix \mathbf{A} , the termination threshold ε .

Initialization: Let the residual vectors $\mathbf{r}_l = \mathbf{y}_l$ where $l = 1, 2, \dots, L$, the support set $\Lambda = \emptyset$ and the number of iteration $t = 0$.

Iteration:

(1) Select $\tilde{i} = \arg \max_{j=1,2,\dots,N} \sum_{l=1}^L |\mathbf{A}_j^H \mathbf{r}_l|$ and let $\Lambda = \Lambda \cup \{\tilde{i}\}$.

(2) Update the residual vector $\mathbf{r}_l = \mathbf{y}_l - \mathbf{A}_\Lambda \mathbf{A}_\Lambda^\dagger \mathbf{y}_l$ where $l = 1, 2, \dots, L$.

(3) $t = t + 1$. If $\sum_{l=1}^L \|\mathbf{r}_l\|_2 < \varepsilon$, terminate the iteration; otherwise, go back to step (1).

Output: Compute the estimated signals $\hat{\mathbf{x}}_i$ by (5).

ALGORITHM 1: SOMP.

Input: The measurement matrix \mathbf{A} , the measurement vectors \mathbf{y}_l where $l = 1, 2, \dots, L$, the sparsity K .

Initialization: Let the residual vectors $\mathbf{r}_{l,old} = \mathbf{y}_l$ where $l = 1, 2, \dots, L$ and the support set $\Lambda_{old} = \emptyset$.

Iteration:

(1) Let the temporary support set $\Lambda_{temp} = \Lambda_{old} \cup \max_indices(\sum_{l=1}^L \text{abs}(\mathbf{A}^H \mathbf{r}_{l,old}), K)$.

(2) Let $\Lambda_{new} = \max_indices(\sum_{l=1}^L \text{abs}(\mathbf{A}_{\Lambda_{temp}}^\dagger \mathbf{y}_l), K)$.

(3) The residual vector is updated as

$\mathbf{r}_{l,new} = \mathbf{y}_l - \mathbf{A}_{\Lambda_{new}} \mathbf{A}_{\Lambda_{new}}^\dagger \mathbf{y}_l$ where $l = 1, 2, \dots, L$.

(4) If $\sum_{l=1}^L \|\mathbf{r}_{l,old}\|_2 > \sum_{l=1}^L \|\mathbf{r}_{l,new}\|_2$, let $\Lambda_{old} = \Lambda_{new}$, $\mathbf{r}_{l,old} = \mathbf{r}_{l,new}$ where $l = 1, 2, \dots, L$ and go back to step (1); otherwise, we terminate the iteration and let the final support set $\Lambda = \Lambda_{old}$.

Output: Compute the estimated signals $\hat{\mathbf{x}}_i$ by (5).

ALGORITHM 2: SSP.

SSP has a mechanism to reevaluate the selected indices. If the selected indices are incorrect, they will be removed from the support set in the next iteration. However, because the adjoining atoms of partial Fourier matrix are strongly correlated and SSP selects multiple atoms at once, it has a weak ability in distinguishing adjoining atoms. We observe that SSP performs worse than SOMP in solving the partial Fourier MMV problem.

HMP combines the strengths of SOMP and SSP. On the one hand, HMP selects atoms by individually applying OMP on local residual vectors (Algorithm 4). This ensures the atoms are selected one by one. On the other hand, in the backtracking stage, it utilizes the mechanism of SSP to reevaluate the selected indices, which will remove the wrong selected indices from the support set. However, in this paper, we summarized the drawbacks of HMP. Firstly, our simulation results demonstrate that it does not always perform better than other algorithms; e.g., it performs worse than SOMP in the case of relative few signals. Moreover, it needs

Input: The measurement vectors \mathbf{y}_l where $l = 1, 2, \dots, L$, the measurement matrix \mathbf{A} , the forward step size α , the backward step size β , the terminate threshold ε , the limit of the size of the estimated support set T .

Initialization: Let residual vectors $\mathbf{r}_l = \mathbf{y}_l$ where $l = 1, 2, \dots, L$, the support set $\Lambda_{old} = \emptyset$ and the number of iteration $t = 0$.

Iteration:

(1) Let the temporary support set

$\Lambda_{temp} = \Lambda_{old} \cup \max_indices(\sum_{l=1}^L \text{abs}(\mathbf{A}^H \mathbf{r}_{l,old}), \alpha)$.

(2) Let

$\Lambda_{new} = \Lambda_{temp} - \min_indices(\sum_{l=1}^L \text{abs}(\mathbf{A}_{\Lambda_{temp}}^\dagger \mathbf{y}_l), \beta)$.

(3) Update the residual vector as

$\mathbf{r}_l = \mathbf{y}_l - \mathbf{A}_{\Lambda_{new}} \mathbf{A}_{\Lambda_{new}}^\dagger \mathbf{y}_l$ where $l = 1, 2, \dots, L$.

(4) $t = t + 1$. If $\sum_{l=1}^L \|\mathbf{r}_l\|_2 < \varepsilon$ or $t > T$, terminate the iteration and let the final support set $\Lambda = \Lambda_{old}$; otherwise, let $\Lambda_{old} = \Lambda_{new}$ and go back to step (1).

Output: Compute the estimated signals $\hat{\mathbf{x}}_i$ by (5).

ALGORITHM 3: FBP.

Input: The measurement matrix \mathbf{A} , the measurement vectors \mathbf{y}_l where $l = 1, 2, \dots, L$, the sparsity K .

Initialization: Let the residual vectors $\mathbf{r}_{l,old} = \mathbf{y}_l$ where $l = 1, 2, \dots, L$ and the support set $\Lambda_{old} = \emptyset$.

Iteration:

(1) Let the temporary support

set $\Lambda_{temp} = \Lambda_{old} \cup \max_indices(\sum_{l=1}^L \text{abs}(\sigma_{l,omp}), K)$ where $\sigma_{l,omp} = \text{OMP}(\mathbf{r}_l, \mathbf{A}, K)$ is the recovered signal by applying OMP on the l -th residual vector.

(2) Let $\Lambda_{new} = \max_indices(\sum_{l=1}^L \text{abs}(\mathbf{A}_{\Lambda_{temp}}^\dagger \mathbf{y}_l), K)$.

(3) The residual vector is updated as

$\mathbf{r}_{l,new} = \mathbf{y}_l - \mathbf{A}_{\Lambda_{new}} \mathbf{A}_{\Lambda_{new}}^\dagger \mathbf{y}_l$ where $l = 1, 2, \dots, L$.

(4) If $\sum_{l=1}^L \|\mathbf{r}_{l,old}\|_2 > \sum_{l=1}^L \|\mathbf{r}_{l,new}\|_2$, let $\Lambda_{old} = \Lambda_{new}$, $\mathbf{r}_{l,old} = \mathbf{r}_{l,new}$ and go back to step (1); otherwise, terminate the iteration and let the final support set $\Lambda = \Lambda_{old}$.

Output: Compute the estimated signals $\hat{\mathbf{x}}_i$ by (5).

ALGORITHM 4: HMP.

sparsity as a prior information which is almost impossible to be known in most practical applications. Although G. Li and R. J. Burkholder [17] propose a method to estimate sparsity, it is not always effective. Once the estimated sparsity is smaller than the real sparsity, the recovery process will fail.

Being a two-step algorithm, FBP has mechanism to reevaluate the selected atoms, and it does not need sparsity as a prior information (Algorithm 3). However, in the forward step, it selects multiple atoms at once, then it is still weak in distinguishing adjoining atoms.

Input: The measurement vectors \mathbf{y}_l where $l = 1, 2, \dots, L$, the measurement matrix \mathbf{A} , the forward step size α , the backward step size β , the terminate threshold ε , the limit of the size of the estimated support set T .

Initialization: Let residual vectors $\mathbf{r}_l = \mathbf{y}_l$ where $l = 1, 2, \dots, L$, the support set $\Lambda_{old} = \emptyset$ and the number of iteration $t = 0$.

Iteration:

(1) Let the temporary support set $\Lambda_{temp} = \Lambda_{old} \cup \max_indices(\sum_{l=1}^L abs(\sigma_{l,omp}), \alpha)$ where $\sigma_{l,omp} = \text{OMP}(\mathbf{r}_l, \mathbf{A}, \alpha)$ is the recovered signal by applying the OMP algorithm on the l -th residual vector.

(2) Let

$$\Lambda_{new} = \Lambda_{temp} - \min_indices(\sum_{l=1}^L abs(\mathbf{A}_{\Lambda_{temp}}^\dagger \mathbf{y}_l), \beta).$$

(3) Update the residual vector as

$$\mathbf{r}_l = \mathbf{y}_l - \mathbf{A}_{\Lambda_{new}} \mathbf{A}_{\Lambda_{new}}^\dagger \mathbf{y}_l \text{ where } l = 1, 2, \dots, L.$$

(4) $t = t + 1$. If $\sum_{l=1}^L \|\mathbf{r}_l\|_2 < \varepsilon$ or $t > T$, terminate the iteration and let the final support set $\Lambda = \Lambda_{old}$; otherwise, let $\Lambda_{old} = \Lambda_{new}$ and go back to step (1).

Output: Compute the estimated signals $\hat{\mathbf{x}}_l$ by (5).

ALGORITHM 5: HOFBP.

4.2. Description of HOFBP. Inspired by the GP algorithms described above, we combine the strengths of SOMP and FBP to propose a Hybrid Orthogonal Forward-Backward Pursuit (HOFBP) algorithm.

HOFBP is summarized in Algorithm 5. In the initialization, the residual vectors are initialized as the measurement vectors and the support set is initialized as the empty set. In HOFBP, the support set expands iteratively. In the first step, as in HMP, L recovered signals are obtained by individually applying OMP on L residual vectors, then, α indices corresponding to the largest amplitude entries of the L recovered signals' fusion result are added to the support set. In the second step, L projection coefficients are obtained by individually projecting measurement vectors onto the space spanned by $\mathbf{A}_{\Lambda_{temp}}$, where Λ_{temp} is the support set. Then, β indices which correspond to the minimum entries in the fusion result of the L projection coefficients are removed from the support set. In the third step, the residual vectors are updated. In the fourth step, the termination condition is checked. If the sum of all residual vectors' l_2 -norms is smaller than the termination threshold or the size of the estimated support set exceeds the maximum allowed size of the support set, the iteration process is terminated, the support set obtained in the last iteration is taken as the final support set and the signals are estimated by using the least squares method as stated in (5). We should point out that this is because sparsity is assumed unknown in HOFBP, we take α as sparsity in the OMP algorithm.

The index selection stage of HOFBP utilizes SOMP. Because OMP is a simplified version of SOMP for one signal, then HOFBP also seems closely related to OMP. HMP combines SOMP and SSP, and HOFBP combines SOMP and FBP. Because SSP and FBP are similar to each other, then, HOFBP also seems closely related to HMP. Although HOFBP seems closely related to OMP and HMP, because SOMP and FBP are basic algorithms of HOFBP, we still say that HOFBP is an algorithm combining SOMP and FBP.

The Halting Conditions. The terminate threshold ε should be set as a very small positive number. In the noiseless case, we set $\varepsilon = 10^{-3}$. In noisy cases, it is set as $\varepsilon = 10^{-SNR/20}$, where $SNR = 10 \log_{10} \|\mathbf{Y}\|_F^2 / \|\mathbf{Noise}\|_F^2$ is the signal-to-noise ratio (SNR). Moreover, in case of a failure, in order to avoid too many iterations, T is used to limit the size of the support set. In order to ensure all correct atoms are found before the support set size reaches T , it should be set larger than the real sparsity K . Because K is unknown, T can be set large enough or simply as M . If the sum of all residual vectors' l_2 -norms is smaller than the terminate threshold ε or the number of iterations exceeds the maximum allowed number of iterations, the iteration process is terminated.

The Choice of Forward and Backward Step Size. The forward step size α and backward step size β are two important parameters for HOFBP. The choices are critical to the performances of the algorithm. α should be set larger than 1 and smaller than the measurement number M . According to the research result of the paper [18], $\alpha \in [0.2M, 0.3M]$ leads to the best performance. According to the definition in HOFBP, β should be set smaller than α . The smaller the value of $\alpha - \beta$, the more carefully the indices are selected. When $\beta = \alpha - 1$, HOFBP has the best performance.

4.3. Relations to Other GP Algorithms. We can see that HOFBP has more strengths over other greedy pursuit algorithms. Firstly, compared with SOMP, it has a mechanism to reevaluate the selected indices. Secondly, compared with SSP, it can distinguish adjoining atoms well by individually applying OMP on local residual vectors. Moreover, it does not need sparsity as a prior information. Thirdly, compared with HMP, the support set expands iteratively, which makes it does not need sparsity as a prior information. Lastly, compared with FBP, in the forward step, it selects atoms one by one, then it can distinguish adjoining atoms well.

We noted the existence of two improved versions of FBP: Iterative Forward-Backward Pursuit (IFBP) [40] and Adaptive Forward-Backward Orthogonal Matching Pursuit (AFBOMP) [41]. However, our algorithm is very different from them. IFBP is an algorithm embedding FBP into the iterative framework. The performance improvement is mainly due to the iterative framework. In AFBOMP, an adaptive method of selecting α and β is proposed. By using this method, FBP's performance is improved. Different from them, the main innovation of our algorithm is the atom selection strategy. The innovations of these three algorithms come from different aspects.

5. Computational Complexity Analysis

SOMP and SSP are extended versions of OMP and SP. According to [30], both OMP and SP have the computational complexity of $O(KMN)$. Because there are L measurement vectors in the MMV problem, both SOMP and SSP have computational complexity of $O(KLMN)$. According to [17], the computational complexity of HMP mainly comes from the computations of applying OMP on L local residual vectors. Thus, each iteration of HMP has the computational complexity of $O(KLMN)$. Since HMP performs $O(K)$ iterations, it has the total computational complexity of $O(K^2LMN)$.

In each iteration of HOFBP, the computational complexity is mainly composed of two parts. Part 1 mainly comes from applying OMP on L residual vectors. Because α is taken as sparsity, the operation of applying OMP on each residual vector has the complexity of $O(\alpha MN)$. Therefore, the operation of applying OMP on L residual vectors has the computational complexity of $O(\alpha LMN)$. Part 2 mainly comes from the projections of L measurement vectors on the space corresponding to the support set in step 2 of the iteration. In step 3 of the t -th ($t > 0$) iteration, the size of support set is $t(\alpha - \beta)$, then, the computations of projecting L measurement vectors have the complexity of $O(t^2(\alpha - \beta)^2 ML)$. Generally speaking, $t(\alpha - \beta) < \alpha \ll N$, in comparison with the complexity of part 1, and the complexities of part 2 are marginal. Therefore, each iteration of HOFBP has the computational complexity of $O(\alpha LMN)$. HOFBP has a maximum of $T/(\alpha - \beta)$ iterations, then the total computational complexity of HOFBP is limited by $O(T\alpha LMN/(\alpha - \beta))$.

6. Simulation Results and Analysis

In this part, we conduct some numerical simulations to evaluate the performance and efficiency of HOFBP. The performance and efficiency of HOFBP are compared with those of HMP, SSP, SOMP, IFBP, and AFBOMP. SOMP, SSP, and HMP are traditional GP algorithms. IFBP and AFBOMP are improved versions of FBP. For IFBP, AFBOMP, and HOFBP, the forward step size α is set as $0.25M$. Noisy and noiseless cases are both considered.

We use the following hypothesis in the simulations.

- (I) The measurement matrix \mathbf{A} is constructed by randomly selecting M rows from a discrete Fourier matrix which has the size of $N \times N$. Every column of \mathbf{A} is normalized to unit l_2 norm.
- (II) Randomly generate L signals. They have same nonzero elements positions. K indices of nonzero elements positions are uniformly chosen from the set $S = \{1, 2, \dots, N\}$. The nonzero elements of each signal are independently chosen from the standard normal distribution.
- (III) All curves in the following figures are obtained by averaging the results over 200 independent trials. In each trial, the signals and measurement matrix \mathbf{A} are generated independently.

- (IV) Perfect recovery percentages are used to evaluate all algorithms' performances. In the i -th trial, if $\|\mathbf{X}^{(i)} - \widehat{\mathbf{X}}^{(i)}\|_2 < 0.05\|\mathbf{X}^{(i)}\|_2$, where $\mathbf{X}^{(i)}$ and $\widehat{\mathbf{X}}^{(i)}$, respectively, denote the original joint signal and the recovered joint signal in the i -th trial, we announce the i -th trial is a perfect recovery. Supposing there is SE perfect recovery in 200 trials, the perfect recovery percentage is $SE/200$.
- (V) Average run times are used to evaluate all algorithms' efficiencies. They are computed by averaging the run times of 200 independent trials.
- (VI) The parameters of algorithms are set as follows: the limit of the size of the estimated support set in HOFBP, IFBP, and AFBOMP is set as $T = M$; the termination threshold of HOFBP, SOMP, IFBP, and AFBOMP is set as $\varepsilon = 10^{-3}$; the stop conditions of SSP and HMP are the real sparsity K .
- (VII) The simulations are performed in Matlab R2014a running on a computer with 3.7 GHz, Intel Xeon E3 processor, with 16.0 GB of RAM, and Windows 7 system.

6.1. Robustness against the Step Size. Firstly, we analyze the influence of the value of step size $\alpha - \beta$ on the performances and efficiencies of HOFBP, IFBP, and AFBOMP. Figure 1 shows the perfect recovery percentages and average run times of these three algorithms against the sparsity K changing from 20 to 38 in the noiseless case. We fix other parameters as $L = 6$, $N = 200$, and $M = 40$. $\alpha - \beta$ varies in $\{1, 3, 5, 7\}$.

For these three algorithms, we can see that $\alpha - \beta = 1$ leads to the best performances. With the increase of the value of $\alpha - \beta$, the performances of these three algorithms decline. That is because the smaller the value of $\alpha - \beta$, the more carefully the indices are selected. For the efficiencies, we can see that the smaller the value of $\alpha - \beta$ is, the more time these three algorithms consume. The reason is that the smaller the value of $\alpha - \beta$ is, the more iterations are needed for the algorithms to reach convergence.

We can see that, for the same $\alpha - \beta$, HOFBP has the highest recovery percentage. For efficiency, HOFBP consumes more time than AFBOMP, however, much less time than IFBP.

6.2. Robustness against Sparsity. Figure 2 shows the perfect recovery percentages and average run times of all algorithms against the sparsity K changing from 20 to 38 in the noiseless case. We fix other parameters as $L = 6$, $N = 200$, and $M = 40$. HMP performs better than SOMP at the beginning. With the increase of sparsity, HMP begins to perform worse than SOMP at the sparsity 24. SSP performs nearly consistently worse than SOMP, which is consistent with our previous analysis. HOFBP performs better than the other algorithms. IFBP has similar performance with HOFBP; however, it consumes much more time than HOFBP. For efficiency, HOFBP consumes similar time to HMP and AFBOMP but performs better than them.

6.3. Robustness against the Measurement Number. Figure 3 shows the perfect recovery percentages and average run

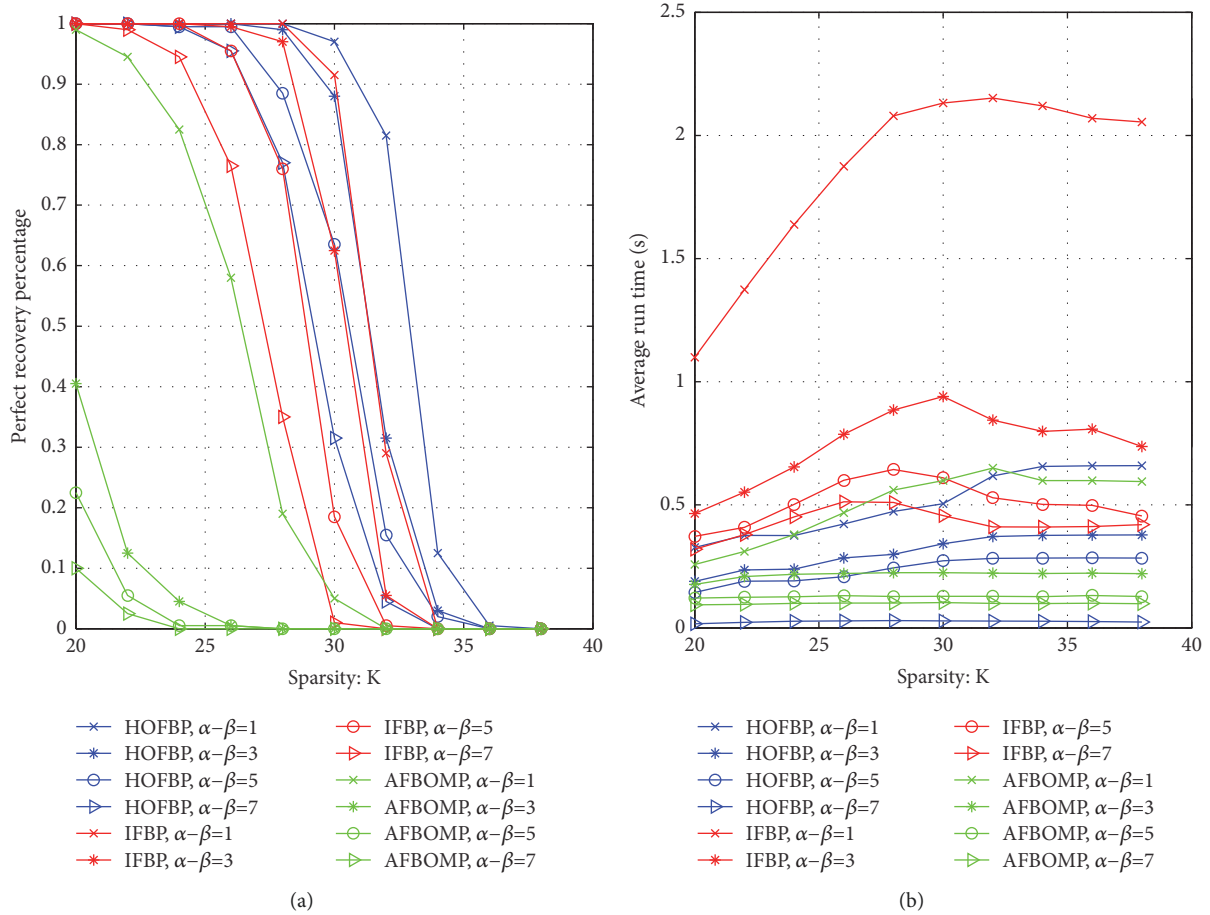


FIGURE 1: Perfect recovery percentages and average run times of HOFBP, IFBP, and AFBOMP with different $\alpha - \beta$ against K with $M = 40$, $N = 200$, $L = 6$.

times of all algorithms against the measurement number M changing from 34 to 60 in the noiseless case. We fix other parameters as $L = 6$, $N = 200$, and $K = 30$. With the increase of the measurement number, the perfect recovery percentages of all algorithms increase. HMP and SSP perform consistently worse than SOMP. HOFBP has the best performance among all algorithms. IFBP performs similarly to HOFBP; however, it consumes much more time. For efficiency, HOFBP is similar to AFBOMP and HMP, but with better performance. Moreover, HOFBP needs the least measurement number to reach perfect recovery.

6.4. Robustness against the Number of Signals. Figure 4 shows the perfect recovery percentages and average run times of all algorithms against the number of signals L changing from 1 to 25 in the noiseless case. We fix other parameters as $M = 45$, $N = 200$, $K = 30$. With the increase of L , the perfect recovery percentages of all algorithms increase. This is because the more measurement vectors are, the more correlations the algorithms can utilize. We can see that HMP performs worse than SOMP when $L < 11$. This illustrates that HMP has less strengths than SOMP in the case of relative less signals. HOFBP performs better than other algorithms. Moreover, it needs less number of signals to reach perfect recovery. For

efficiency, we have similar results to those of Sections 4.2 and 4.3.

6.5. Recovery from Noisy Observation. In this part, we test algorithms' performances against noise. We assume the noise is i.i.d Gaussian noise. The signal-to-noise ratio (SNR) is used to measure the relation between the measurement signal and the noise, which is defined as

$$\text{SNR} = \frac{10 \log_{10} \|\mathbf{Y}\|_F^2}{\|\text{Noise}\|_F^2}. \quad (8)$$

What should be pointed out is that, in the noisy case, the terminate parameters of SOMP, HOFBP, IFBP, and AFBOMP are set as $\varepsilon = 10^{-\text{SNR}/20}$. The terminate parameters of SSP and HMP are still set as the real sparsity.

Figure 5 shows the perfect recovery percentages and average run times of all algorithms against the change of noise. Other parameters are fixed as $M = 45$, $K = 30$, $N = 200$, and $L = 6$. With the increase of SNR, the perfect recovery percentages of all algorithms increase. SSP and HMP perform consistently worse than SOMP. AFBOMP performs worst, which means it is not suitable for the noisy situation. HOFBP performs consistently better than other

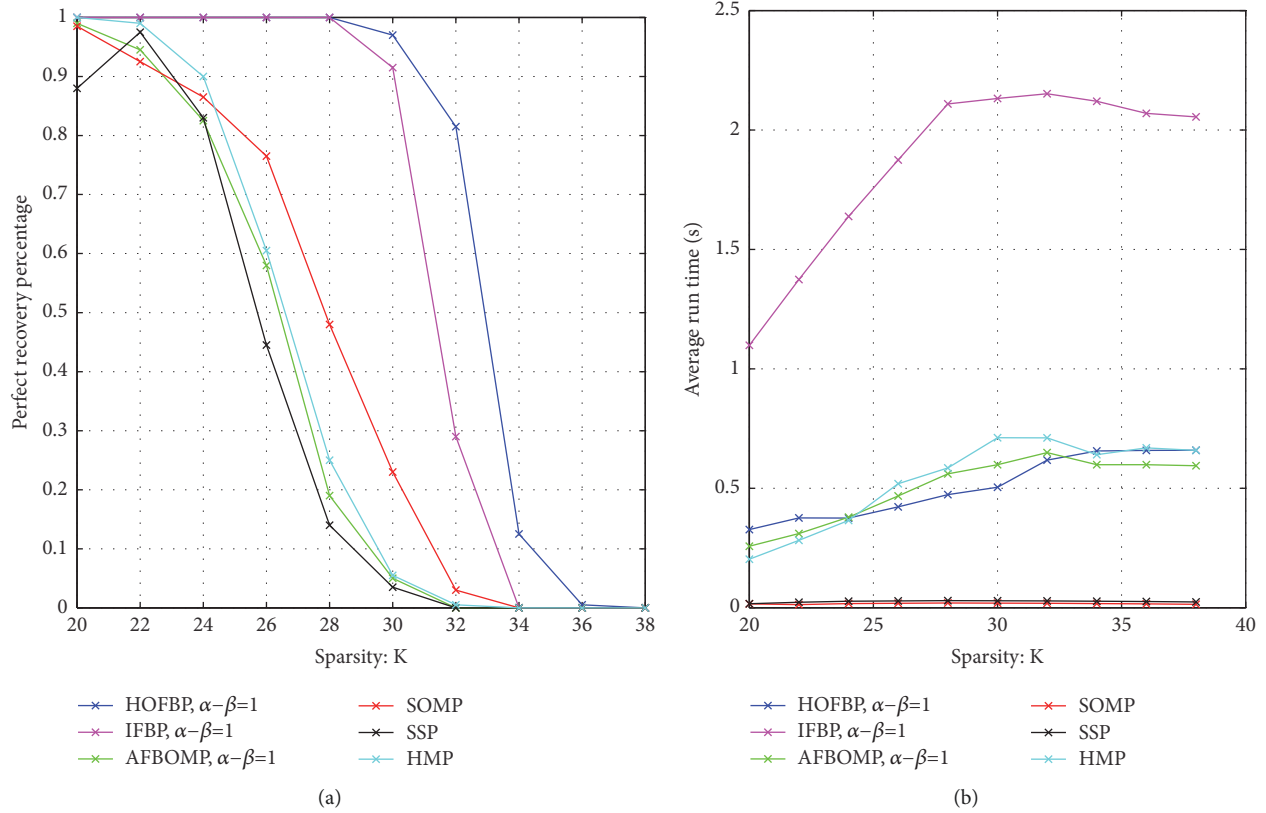


FIGURE 2: Perfect recovery percentages and average run times of different algorithms against K with $M = 40, N = 200, L = 6$.

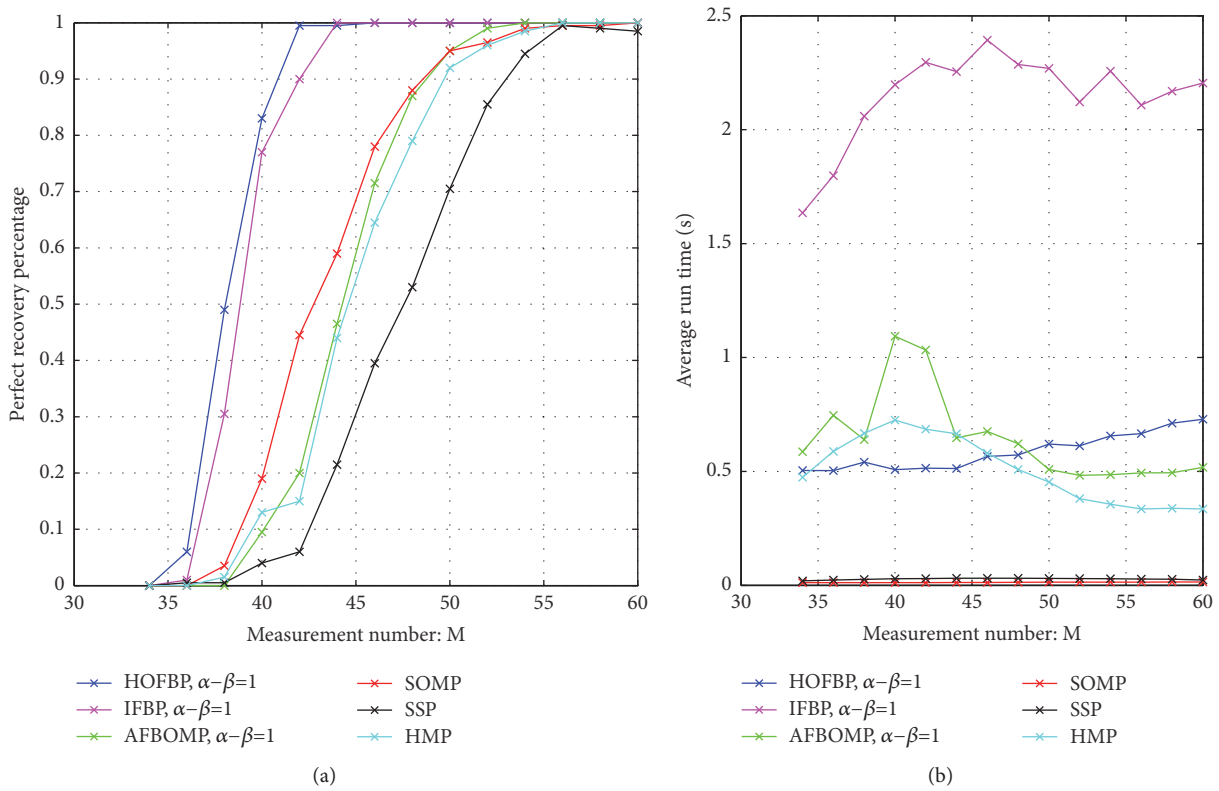


FIGURE 3: Perfect recovery percentages and average run times of different algorithms against M with $K = 30, N = 200, L = 6$.

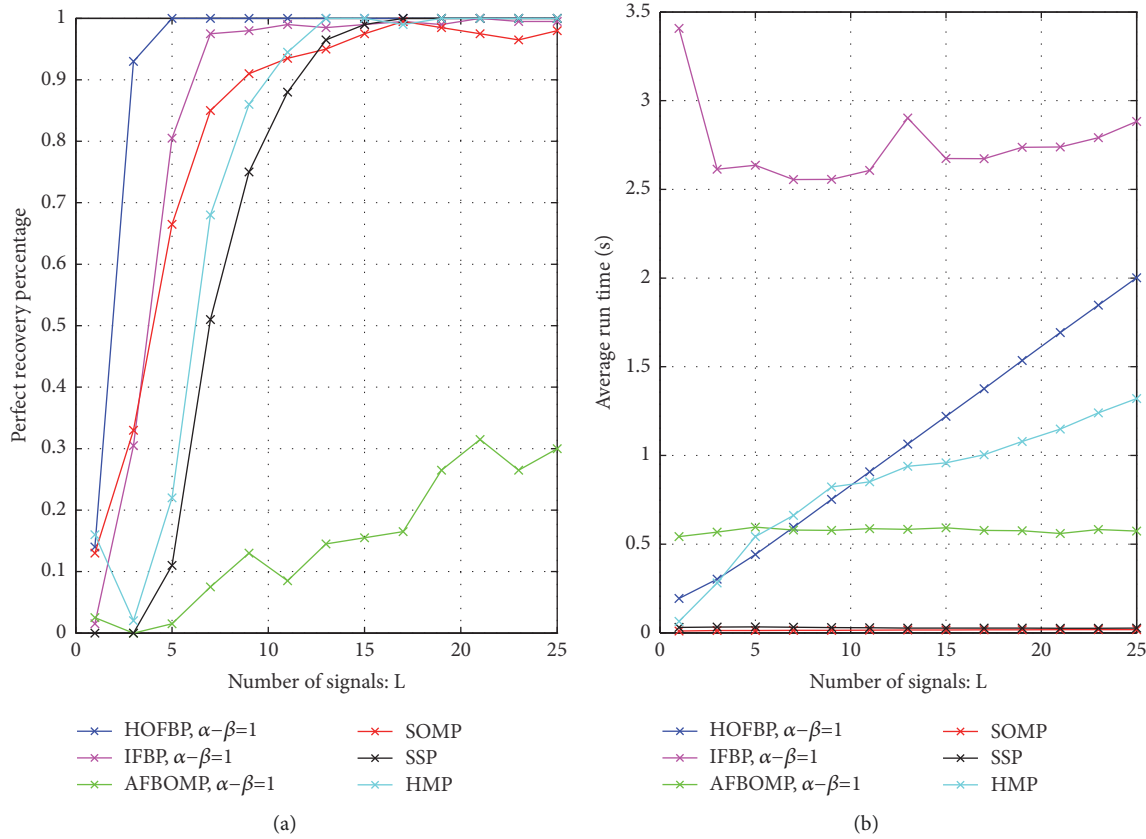


FIGURE 4: Perfect recovery percentages and average run times of different algorithms against L with $M = 45, K = 30, N = 200$.

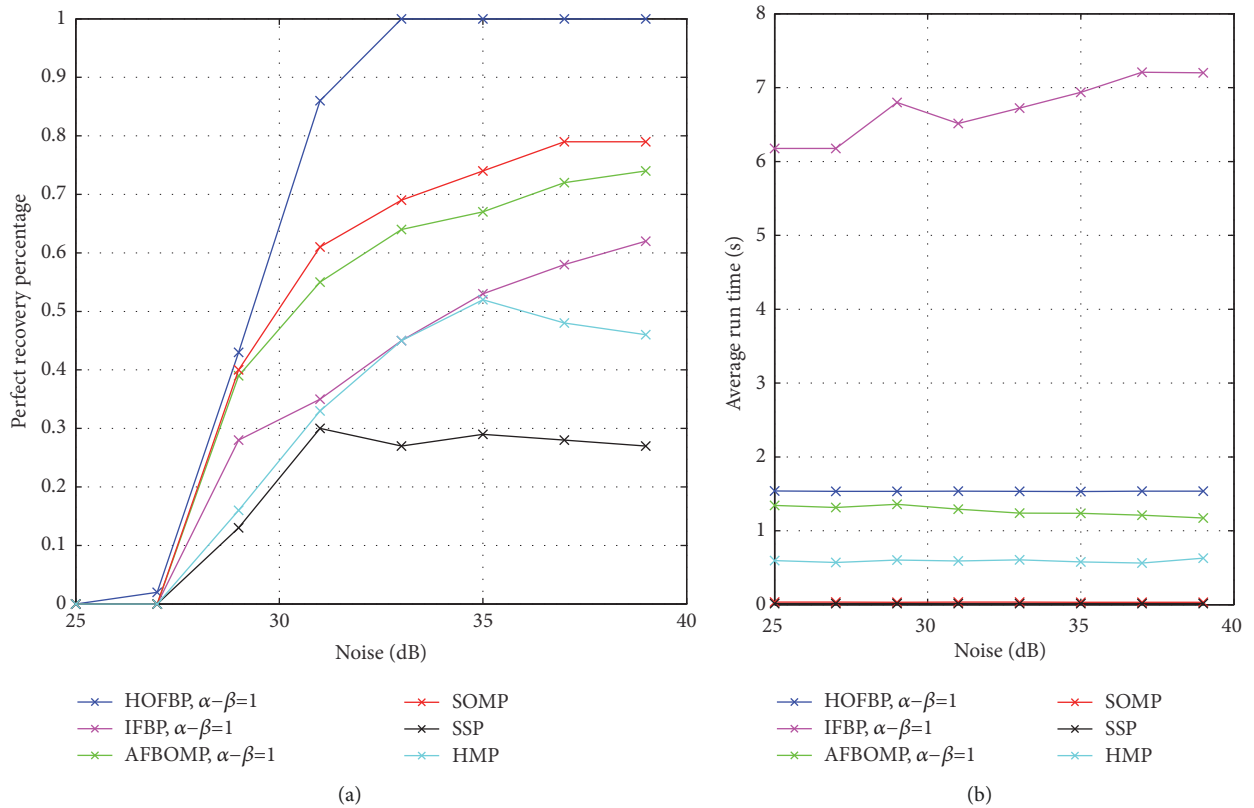


FIGURE 5: Perfect recovery percentages and average run times of different algorithms against SNR with $M = 45, K = 30, N = 200$, and $L = 6$.

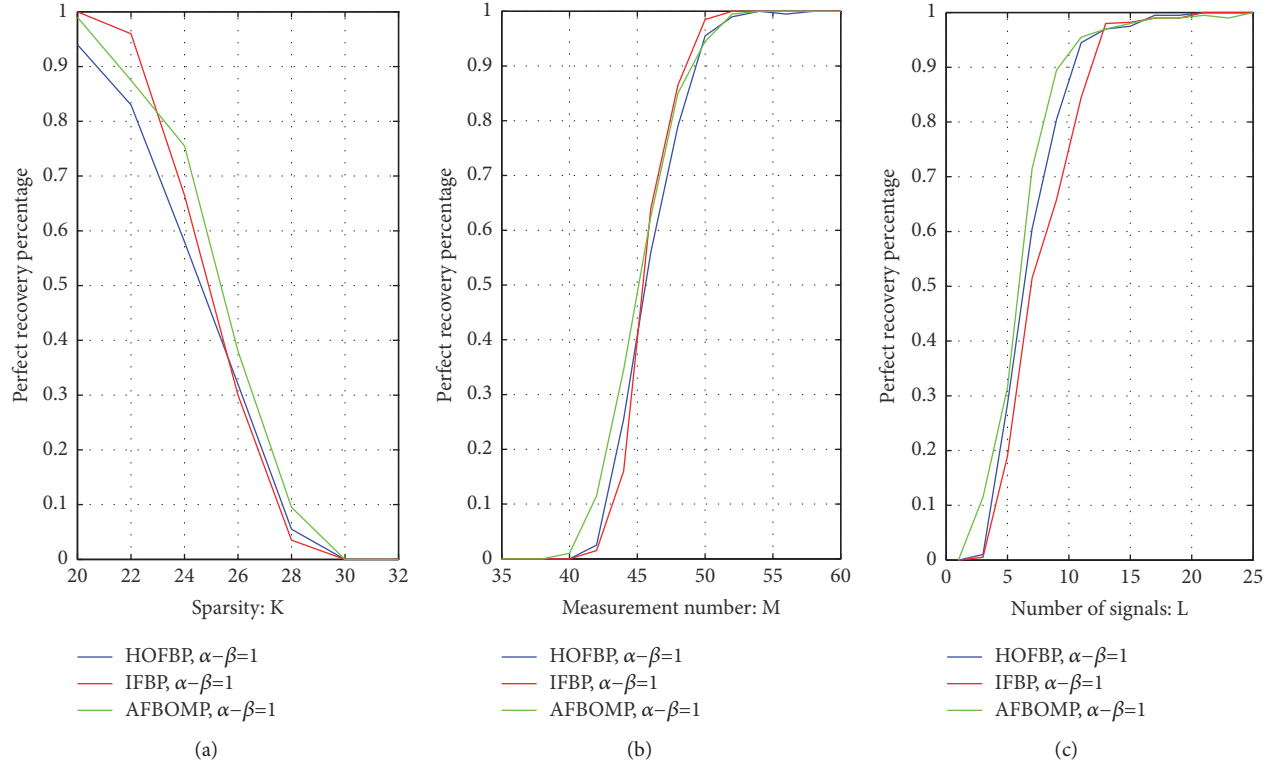


FIGURE 6: Perfect recovery percentages of HOFBP, IFBP, and AFBOMP (a) against K with $M = 40$, $N = 200$, $L = 6$; (b) against M with $K = 30$, $N = 200$, $L = 6$; (c) against L with $M = 45$, $K = 30$, $N = 200$.

algorithms. For efficiency, we have similar result to those of Sections 4.2 and 4.3.

7. Conclusion

The strengths and drawbacks of previous GP algorithms are analyzed. On this basis, we combine SOMP and FBP to propose the HOFBP algorithm. The choices of the parameters are critical to the performance of this algorithm. However, the simulation results show that, by using proper parameters, HOFBP has better performance than other GP algorithms at the expense of more computing time in some cases. Moreover, HOFBP does not need sparsity as a prior information. It is important to give the exact expressions for the mean square error in the construction using our algorithm. This will be researched in our future work. Moreover, in my opinion, AFBOMP, IFBP, and HOFBP can be combined to improve the performance further. This also can be researched in the future work.

Appendix

As illustrated in Section 2.3, Gaussian random measurement matrix is very different from partial Fourier measurement matrix. The adjoining atoms of Gaussian random measurement matrix are not highly correlated to each other. In order to demonstrate this point, we do some experiments.

IFBP, AFBOMP, and our algorithm are improved versions of FBP. In Section 6, our algorithm shows better performance

than IFBP and AFBOMP in solving partial Fourier MMV problem. In this part, our algorithm is compared with them in solving the Gaussian random MMV problem.

The experimental design is the same as in Section 6, except the measurement matrix has Gaussian random entries.

Figures 6(a), 6(b), and 6(c) individually show the performances of these three algorithms against the change of sparsity K , measurement number M , and the number of signals L . It can be seen that our algorithm has not advantage over IFBP and AFBOMP in solving a Gaussian random MMV problem.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was supported by National Natural Science Foundation of China (no. 51574232 and no. 61379100).

References

- [1] D. L. Donoho, "Compressed sensing," *IEEE Transactions on Information Theory*, vol. 52, no. 4, pp. 1289–1306, 2006.
- [2] E. J. Candes, J. Romberg, and T. Tao, "Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information," *Institute of Electrical and Electronics*

- Engineers Transactions on Information Theory*, vol. 52, no. 2, pp. 489–509, 2006.
- [3] E. J. Candes and T. Tao, “Near-optimal signal recovery from random projections: universal encoding strategies?” *Institute of Electrical and Electronics Engineers Transactions on Information Theory*, vol. 52, no. 12, pp. 5406–5425, 2006.
 - [4] S. F. Cotter, B. D. Rao, K. Engan, and K. Kreutz-Delgado, “Sparse solutions to linear inverse problems with multiple measurement vectors,” *IEEE Transactions on Signal Processing*, vol. 53, no. 7, pp. 2477–2488, 2005.
 - [5] C. Qi, X. Wang, and L. Wu, “Underwater acoustic channel estimation based on sparse recovery algorithms,” *IET Signal Processing*, vol. 5, no. 8, pp. 739–747, 2011.
 - [6] D. Malioutov, M. Cetin, and A. S. Willsky, “A sparse signal reconstruction perspective for source localization with sensor arrays,” *IEEE Transactions on Signal Processing*, vol. 53, no. 8, pp. 3010–3022, 2005.
 - [7] A. C. Gurbuz, V. Cevher, and J. H. McClellan, “Bearing estimation via spatial sparsity using compressive sensing,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 48, no. 2, pp. 1358–1369, 2012.
 - [8] H. P. Babcock, J. R. Moffitt, Y. Cao, and X. Zhuang, “Fast compressed sensing analysis for superresolution imaging using L1-homotopy,” *Optics Express*, vol. 21, no. 23, pp. 28583–28596, 2013.
 - [9] A. Morelli Andrés, S. Padovani, M. Tepper, and J. Jacobo-Berlles, “Face recognition on partially occluded images using compressed sensing,” *Pattern Recognition Letters*, vol. 36, no. 1, pp. 235–242, 2014.
 - [10] S.-J. Wei, X.-L. Zhang, J. Shi, and G. Xiang, “Sparse reconstruction for SAR imaging based on compressed sensing,” *Progress in Electromagnetics Research*, vol. 109, pp. 63–81, 2010.
 - [11] R. O. Schmidt, “Multiple emitter location and signal parameter estimation,” *IEEE Transactions on Antennas and Propagation*, vol. 34, no. 3, pp. 276–280, 1986.
 - [12] Z. Wen, B. Hou, and L. Jiao, “Joint sparse recovery with semisupervised music,” *IEEE Signal Processing Letters*, vol. 24, no. 5, pp. 629–633, 2017.
 - [13] J. M. Kim, O. K. Lee, and J. C. Ye, “Compressive MUSIC: revisiting the link between compressive sensing and array signal processing,” *IEEE Transactions on Information Theory*, vol. 58, no. 1, pp. 278–301, 2012.
 - [14] D. P. Wipf and B. D. Rao, “An empirical Bayesian strategy for solving the simultaneous sparse approximation problem,” *IEEE Transactions on Signal Processing*, vol. 55, no. 7, part 2, pp. 3704–3716, 2007.
 - [15] Z. Zhang and B. D. Rao, “Sparse signal recovery with temporally correlated source vectors using sparse Bayesian learning,” *IEEE Journal of Selected Topics in Signal Processing*, vol. 5, no. 5, pp. 912–926, 2011.
 - [16] J. Ziniel and P. Schniter, “Efficient high-dimensional inference in the multiple measurement vector problem,” *IEEE Transactions on Signal Processing*, vol. 61, no. 2, pp. 340–354, 2013.
 - [17] G. Li and R. J. Burkholder, “Hybrid matching pursuit for distributed through-wall radar imaging,” *Institute of Electrical and Electronics Engineers. Transactions on Antennas and Propagation*, vol. 63, no. 4, part 2, pp. 1701–1711, 2015.
 - [18] J. A. Tropp, A. C. Gilbert, and M. J. Strauss, “Algorithms for simultaneous sparse approximation. Part I: greedy pursuit,” *Signal Processing*, vol. 86, no. 3, pp. 572–588, 2006.
 - [19] J.-M. Feng and C.-H. Lee, “Generalized subspace pursuit for signal recovery from multiple-measurement vectors,” in *Proceedings of the 2013 IEEE Wireless Communications and Networking Conference (WCNC '13)*, pp. 2874–2878, April 2013.
 - [20] J. D. Blanchard, M. Cermak, D. Hanle, and Y. Jing, “Greedy algorithms for joint sparse recovery,” *IEEE Transactions on Signal Processing*, vol. 62, no. 7, pp. 1694–1704, 2014.
 - [21] R. Gribonval, H. Rauhut, K. Schnass, and P. Vandergheynst, “Atoms of all channels, unite! Average case analysis of multi-channel sparse recovery using greedy algorithms,” *Journal of Fourier Analysis and Applications*, vol. 14, no. 5-6, pp. 655–687, 2008.
 - [22] P. Maechler, N. Felber, and H. Kaeslin, “Compressive sensing for WiFi-based passive bistatic radar,” in *Proceedings of the 20th European Signal Processing Conference (EUSIPCO '12)*, pp. 1444–1448, August 2012.
 - [23] N. B. Karahanoglu and H. Erdogan, “Compressed sensing signal recovery via forward-backward pursuit,” *Digital Signal Processing*, vol. 23, no. 5, pp. 1539–1548, 2013.
 - [24] L. Stanković, M. Daković, S. Stanković, and I. Orović, *Sparse Signal Reconstruction—Introduction*, Wiley Encyclopedia of Electrical and Electronics Engineering, John Wiley & Sons, Inc, 2017.
 - [25] E. Yang, T. Zhang, X. Yan, Q. Wang, and K. Qin, “Arbitrary block-sparse signal reconstruction based on incomplete single measurement vector,” *Circuits, Systems and Signal Processing*, vol. 36, no. 11, pp. 4569–4592, 2017.
 - [26] E. J. Candes and T. Tao, “Decoding by linear programming,” *IEEE Transactions on Information Theory*, vol. 51, no. 12, pp. 4203–4215, 2005.
 - [27] S. S. Chen, D. L. Donoho, and M. A. Saunders, “Atomic decomposition by basis pursuit,” *SIAM Review*, vol. 43, no. 1, pp. 129–159, 2001.
 - [28] D. L. Donoho and M. Elad, “Optimally sparse representation in general (nonorthogonal) dictionaries via L1 minimization,” *Proceedings of the National Academy of Sciences of the United States of America*, vol. 100, no. 5, pp. 2197–2202, 2003.
 - [29] 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.
 - [30] W. Dai and O. Milenkovic, “Subspace pursuit for compressive sensing signal reconstruction,” *IEEE Transactions on Information Theory*, vol. 55, no. 5, pp. 2230–2249, 2009.
 - [31] 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.
 - [32] D. Needell and R. Vershynin, “Uniform uncertainty principle and signal recovery via regularized orthogonal matching pursuit,” *Foundations of Computational Mathematics*, vol. 9, no. 3, pp. 317–334, 2009.
 - [33] D. L. Donoho, Y. Tsaig, I. Drori, and J. Starck, “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.
 - [34] J. Wang, S. Kwon, and B. Shim, “Generalized orthogonal matching pursuit,” *IEEE Transactions on Signal Processing*, vol. 60, no. 12, pp. 6202–6216, 2012.
 - [35] S. Li, F. Gao, G. Ge, and S. Zhang, “Deterministic construction of compressed sensing matrices via algebraic curves,” *IEEE Transactions on Information Theory*, vol. 58, no. 8, pp. 5035–5041, 2012.

- [36] A. Amini, V. Montazerhodjat, and F. Marvasti, "Matrices with small coherence using p-ary block codes," *IEEE Transactions on Signal Processing*, vol. 60, no. 1, pp. 172–181, 2012.
- [37] F. Wang, J. Zhang, G. Sun et al., "A survey of dimensionality reduction techniques based on random projection," *Pattern Recognition*, 2017.
- [38] D. Achlioptas, "Database-friendly random projections: johnson-lindenstrauss with binary coins," *Journal of Computer & System Sciences*, vol. 66, no. 4, pp. 671–687, 2003.
- [39] P. Li, T. Hastie, and K. W. Church, "Very sparse random projections," in *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '06)*, pp. 287–296, Philadelphia, PA, USA, August 2006.
- [40] F. Wang, J. Zhang, G. Sun, and T. Geng, "Iterative forward-backward pursuit algorithm for compressed sensing," *Journal of Electrical and Computer Engineering*, vol. 2016, Article ID 5940371, 7 pages, 2016.
- [41] N. Mourad, M. Sharkas, and M. M. Elsherbeny, "Adaptive forward-backward orthogonal matching pursuit for compressed sensing," in *Proceedings of the 33rd National Radio Science Conference (NRSC '16)*, pp. 114–121, February 2016.

