

Research Article

Binary Sparse Phase Retrieval via Simulated Annealing

Wei Peng and Hongxia Wang

College of Science, National University of Defense Technology, Changsha, Hunan 410073, China

Correspondence should be addressed to Wei Peng; weipeng0098@126.com

Received 11 March 2016; Revised 23 April 2016; Accepted 4 May 2016

Academic Editor: Pasquale Memmolo

Copyright © 2016 W. Peng and H. Wang. 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.

This paper presents the Simulated Annealing Sparse Phase Recovery (SASPAR) algorithm for reconstructing sparse binary signals from their phaseless magnitudes of the Fourier transform. The greedy strategy version is also proposed for a comparison, which is a parameter-free algorithm. Sufficient numeric simulations indicate that our method is quite effective and suggest the binary model is robust. The SASPAR algorithm seems competitive to the existing methods for its efficiency and high recovery rate even with fewer Fourier measurements.

1. Introduction

Phase retrieval, well known as the problem of recovering a signal from magnitudes of its Fourier transform, has numerous applications including crystallography [1], optical imaging [2], speech processing [3], and astronomical imaging [4].

Reconstruction of a signal from its magnitudes of a linear transform is a difficult task, which is usually formulated as a nonconvex optimization problem. Algorithms for the problem could be divided into the projection type and the lifting type. Projection algorithms like error reduction [5], Fienup [6], and HPR [7] project the signal \mathbf{x} onto two feasible sets alternately, at least one of which is nonconvex. These algorithms have been reported to be variants of their convex counterparts essentially [8]. Though they might work well in engineering applications, they usually lack convergence to the global minima. Phase lifting technique based on matrix completion [9] is developed to transform phase retrieval problems to be convex. However, the technique would square the number of variables for executing semidefinite programming (SDP), which restricts the lifting technique to small scale problems. More algorithms are proposed in recent; see Wirtinger Flow [10] and alternating minimization [11].

With the popularity of compressing sensing, the sparsity of signals also attracts attentions in phase retrieval. For the sparse cases, algorithms on different strategies are presented; see Bayesian Network based message passing [12], greedy strategy based GESPAR [13], projection based sparse-Fienup

[14], lifting based SDP algorithms [15], and more. Analysis of uniqueness for sparse phase retrieval could be found [16, 17].

In the field of compressing sensing, sparse binary signals are also studied [18]. In this paper, we are interested in recovering a binary or boolean vector $\mathbf{x} \in \{0, 1\}^n$ from magnitudes of its complex linear measurements without any phase information. That is, given measurement vectors $\mathbf{a}_i \in \mathbb{C}^n$, suppose that

$$y_i = |\langle \mathbf{a}_i, \mathbf{x} \rangle|^2, \quad \text{for } i = 1, \dots, m; \quad (1)$$

then the goal is to reconstruct \mathbf{x} via \mathbf{y} and the measurement matrix

$$A = [\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_m]^H. \quad (2)$$

The setting of A is various; here are three common types of measurements:

- (1) The most classic type of A is the discrete Fourier matrix. It is usually impossible to recover $1 - D$ signal uniquely regardless of trivial transforms (shifting, mirroring, and phase shifting) without extra information provided. Fortunately, constrained by sparsity, the original signal might be reconstructed uniquely.
- (2) The measurement matrix A could be complex Gaussian matrix inheriting compressive sensing.
- (3) The measurement matrix A could be several masked discrete Fourier matrices, which increases the number of measurements.

The binary signal or boolean signal exists widely, such as the computer system and boolean objects in microscopic imaging systems. However, insufficient studies focus on the binary phase retrieval. In general, binary phase retrieval belongs to pseudo-boolean optimization, which is written as

$$\begin{aligned} \min \quad & f(\mathbf{x}) \\ \text{s.t.} \quad & x_i \in \{-1, 1\}, \end{aligned} \quad (3)$$

where f maps $\{-1, 1\}^n$ to \mathbb{R} .

Simulated annealing (SA) [19], as a probabilistic method to approximate the global optimum of the objective function, is often considered when the feasible set is discrete. Moreover, simulated annealing has been employed in the previous classic phase retrieval studies. Reference [20] performs simulated annealing on the level-quantized phase variables for front-wave imaging while [21] also applies SA to the phase variables. Reference [22] introduces SA for finding periodic phase relief structures via phase perturbations. In [23, 24], the SA methods are also applied to the phase retrieval problems but only require 0-order information without gradients. However, for the recently prevailing sparse phase retrieval, it seems that simulated annealing is still waiting to be exploited. Therefore, simulated annealing for Sparse Phase Retrieval (SASPAR) is presented in this paper. We will identify the nonzeros in signals via simulated annealing based on the local gradient. The numerical experiments indicate the outstanding performance of SASPAR to recover signals from magnitudes of their Fourier transform. The recovery rate even outperforms GESPAR, which is the state-of-the-art algorithm for sparse phase retrieval. Especially in the cases with fewer measurements, our algorithm could also recover the signal with a high probability.

Notations and formulations for the binary sparse phase retrieval are given in Section 2. Algorithms are presented in Section 3. Several numerical simulations are conducted in Section 4 and Section 5 concludes the paper.

2. Notation and Setup

The matrices will be represented by upper letters. Vectors and vector-valued functions are denoted by lower bold letters like \mathbf{x} and $\mathbf{p}(i)$. The i th element of a vector \mathbf{x} will be denoted by x_i . The conjugate transpose of a matrix A will be denoted by A^H . Pointwise multiplication will be represented by \odot . The gradient of a real-valued function $F(\mathbf{x})$ will be denoted by $\nabla F(\mathbf{x})$, the i th element of which is denoted by $\nabla_i F(\mathbf{x})$. The square root of a vector $\mathbf{y} \in \mathbb{R}^m$ is defined pointwisely; that is, $\sqrt{\mathbf{y}} = [\sqrt{y_1}, \sqrt{y_2}, \dots, \sqrt{y_m}]^T$. $|\cdot|$ with a set S in it denoting the total number of elements in S . $\{0, 1\}$ is also denoted by \mathbb{B} . The sparsity $\|\mathbf{x}\|_0$ equals the number of nonzeros in \mathbf{x} .

A general version of sparse phase retrieval is

$$\begin{aligned} \min \quad & F(\mathbf{x}) = \sum_{i=1}^m \left(|\mathbf{a}_i^H \mathbf{x}|^2 - y_i \right)^2 \\ \text{s.t.} \quad & \|\mathbf{x}\|_0 \leq k, \quad (0 < k \ll n) \\ & \mathbf{x} \in \mathbb{R}^n, \end{aligned} \quad (4)$$

where $\mathbf{a}_i \in \mathbb{C}^n$ is a complex measurement vector. The problem is essentially a quadratic optimization problem.

Estimation of Sparsity k . For a unitary linear transform U , including the Fourier transform, the sparsity is obvious, which could be estimated accurately by

$$k = \|\mathbf{x}\|_2^2 = \|U\mathbf{x}\|_2^2 = \|\sqrt{\mathbf{y}}\|_2^2 = \sum_{i=1}^m y_i. \quad (5)$$

For complex Gaussian matrix G , the sparsity could be estimated by $\bar{k} = \sum_{i=1}^m y_i / 2m$ (see Appendix). In our work, $k = \|\mathbf{x}\|_0$ is assumed to be given as the prior information.

Denote all the possible indexes where x_i might be one as I , which is also called the support of \mathbf{x} . In Fourier cases, zeros are added to the end of signals for obtaining high frequency Fourier measurements (see details in [13]), which is called oversampling.

In this paper, as we restrict the signal \mathbf{x} to be binary sparse and the sparsity k is already known, then problem (4) is formulated as

$$\begin{aligned} \min \quad & F(\mathbf{x}) = \sum_{i=1}^m \left(|\mathbf{a}_i^H \mathbf{x}|^2 - y_i \right)^2 \\ \text{s.t.} \quad & \|\mathbf{x}\|_0 = k, \quad (0 \leq k \ll n) \\ & \mathbf{x} \in \mathbb{B}^n, \\ & \{i \mid x_i \neq 0\} \subset I, \end{aligned} \quad (6)$$

where $y_i \geq 0$ ($\forall 0 \leq i \leq m$). It could also be formulated in the matrix form,

$$\begin{aligned} \min \quad & F(\mathbf{x}) = \|\mathbf{A}\mathbf{x} \odot |\mathbf{A}\mathbf{x}| - \mathbf{y}\|_2^2 \\ \text{s.t.} \quad & \|\mathbf{x}\|_0 = k, \quad (0 < k \ll n) \\ & \mathbf{x} \in \mathbb{B}^n, \\ & \{i \mid x_i \neq 0\} \subset I, \end{aligned} \quad (7)$$

where $A = [\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_m]^H$ and $\mathbf{y} = [y_1, y_2, \dots, y_m]^T$.

3. Algorithms

As the sparsity $\|\mathbf{x}\|_0 = k$ is assumed to be given, the problem is to identify the locations of k ones in \mathbf{x} , which is a combinatorial optimization problem. Size of the feasible set is

$$|\{x \in \mathbb{B}^n \mid \|\mathbf{x}\|_0 = k\}| = \binom{n}{k} \approx \mathcal{O}(n^k), \quad (n \gg k). \quad (8)$$

The size would be quite huge. For example, let $n = 1024$ and $k = 30$. Then it will cost about 10^{90} evaluations to search the whole feasible set, which seems impossible for an ordinary computer at present. Simulated annealing is an effective tool for this type of problems. Algorithm 1 gives

```

Require:  $T_0, T_{\text{end}}, 0 < \alpha < 1, \mathbf{x}_0$ 
Initialization:  $\mathbf{x}_{\text{new}} \leftarrow \mathbf{x}_0, T \leftarrow T_0, \text{iter} \leftarrow 0$ 
while  $T \geq T_{\text{end}}$  do
   $\text{iter} \leftarrow \text{iter} + 1$ 
   $T \leftarrow \alpha T$ 
  Select a random neighbour from  $\mathcal{N}(\mathbf{x}_{\text{new}})$  according to probability the  $\mathbb{P}$  defined on  $\mathcal{N}(\mathbf{x}_{\text{new}})$ 
   $\Delta \leftarrow F(\mathbf{x}_{\text{new}}) - F(\mathbf{x}_{\text{iter}})$ 
  if  $\Delta \leq 0$  then
     $\mathbf{x}_{\text{iter}+1} \leftarrow \mathbf{x}_{\text{new}}$ 
  else
    if  $\exp(-\Delta/T) > \text{rand}(1)$  then
       $\mathbf{x}_{\text{iter}+1} \leftarrow \mathbf{x}_{\text{new}}$ 
    else
       $\mathbf{x}_{\text{iter}+1} \leftarrow \mathbf{x}_{\text{iter}}$ 
       $\mathbf{x}_{\text{new}} \leftarrow \mathbf{x}_{\text{iter}}$ 
    end if
  end if
end while
return  $\mathbf{x}_{\text{opt}} = \arg \min_{0 \leq \text{iter} \leq \text{maxIter}} F(\mathbf{x}_{\text{iter}})$ 

```

ALGORITHM 1: Simulated annealing.

a general frame of simulated annealing. The details of the frame will be discussed.

Cooling Schedule. An exponential cooling schedule is employed [25]. That is, the temperature T in the iter-th iteration is defined as

$$T(\text{iter}) = T_0 \alpha^{\text{iter}}, \quad (9)$$

where $0 < \alpha < 1$.

Neighbourhood. The neighbourhood is defined to construct the search graph. Suppose we already know \mathbf{x}_{iter} and define a set map $S(\mathbf{x}) = \{i \mid x_i = 1\}$. First of all, $\nabla F(\mathbf{x}_{\text{iter}})$ is computed (see computation details in Appendix B). In particular, the gradient for the discrete Fourier transform matrix A is

$$\nabla F(\mathbf{x}) = 4A^H \left[(|\mathbf{Ax}|^2 - \mathbf{y}) \odot \mathbf{Ax} \right], \quad (10)$$

where \odot denotes pointwise multiplication. Let the index set be

$$I_1 = \{i \mid \nabla_i F(\mathbf{x}_{\text{iter}}) > 0\} \cap S(\mathbf{x}_{\text{iter}}). \quad (11)$$

I_1 consists of the locations where $x_i = 1$ with positive gradient. Then define neighbours of \mathbf{x}_{iter} as

$$\begin{aligned} \mathcal{N}(\mathbf{x}_{\text{iter}}) \\ = \{S(\mathbf{x}_{\text{iter}}) \setminus \{i_1\} \cup \{i_2\} \mid i_1 \in I_1, i_2 \in S^C(\mathbf{x}_{\text{iter}})\}. \end{aligned} \quad (12)$$

Neighbours of \mathbf{x}_{iter} are the ones which are k sparsity and swap one pair between $S(\mathbf{x}_{\text{iter}})$ and $S^C(\mathbf{x}_{\text{iter}})$.

Randomized Selection. In order to avoid trapping by a local minima in a single loop, we select a neighbour randomly. The discrete distribution defined on $\mathcal{N}(\mathbf{x}_{\text{iter}})$ is presented in two stages.

Stage 1. First, $i_1 \in I_1$ is selected according to the discrete probability density distribution on I_1 . We define the distribution function

$$p_1(i) = \frac{\nabla_i F(\mathbf{x}_{\text{iter}})}{\sum_{i \in I_1} \nabla_i F(\mathbf{x}_{\text{iter}})}, \quad i \in I_1. \quad (13)$$

The index with larger positive gradient tends to be selected.

Stage 2. After i_1 is determined, x_{i_1} is set to 0 and a $k - 1$ sparsity vector \mathbf{x}_{mid} is obtained. Define the index set

$$I_2 = \{i \in I \mid \nabla_i F(\mathbf{x}_{\text{mid}}) < 0\} \cap S^C(\mathbf{x}_{\text{iter}}). \quad (14)$$

Since $I_2 \subset S^C(\mathbf{x}_{\text{iter}})$, $\forall i_2 \in I_2$, setting the i_2 th element of \mathbf{x}_{mid} to 1 could derive a k sparsity vector $\mathbf{x}_{\text{new}} \in \mathcal{N}(\mathbf{x}_{\text{iter}})$. The greedy strategy is employed in this step. The index with smallest negative gradient is selected; that is,

$$i_2 = \arg \min_{i \in I_2} \nabla_i F(\mathbf{x}_{\text{mid}}). \quad (15)$$

Combined with the specification of the search graph and randomized selection, the Simulated Annealing Sparse Phase Retrieval (SASPAR) for binary signals is presented as Algorithm 2 displays.

A more naive and straight algorithm of searching the minima is also presented. Without probabilistic selection of the swapped indexes, simply greedy 2-opt [26] strategy is employed in Algorithm 3, that is, exchanging the index with positive maximal gradient in I_1 and the index with negative minimal gradient in I_2 . Greedy 2-opt binary phase retrieval is an algorithm which is easy to implement and parameter-free. The comparison is conducted in the simulation part.

4. Numeric Simulation

In order to illustrate the performance of SASPAR, several experiments are conducted. To our knowledge, there are few

```

Require:  $T_0, T_{\text{end}}, \alpha, \mathbf{x}_0$ 
Initialization:  $\mathbf{x}_{\text{new}} \leftarrow \mathbf{x}_0, T \leftarrow T_0, \text{iter} \leftarrow 0$ 
while  $T \geq T_{\text{end}}$  do
   $\text{iter} \leftarrow \text{iter} + 1$ 
   $T \leftarrow \alpha T$ 
  Obtain  $I_1 = \{i \in I \mid \nabla_i F(\mathbf{x}_{\text{iter}}) > 0 \cap S(\mathbf{x}_{\text{iter}})\}$ 
  Select  $i_1$  with the probability:
    
$$\mathbb{P}(i_1 = i) = \frac{\nabla_i F(\mathbf{x}_{\text{iter}})}{\sum_{i \in I_1} \nabla_i F(\mathbf{x}_{\text{iter}})}$$

   $\mathbf{x}_{\text{new}}(i_1) \leftarrow 0$ 
  Select  $i_2$  such that
    
$$i_2 = \arg \min_{i_2 \in S^c(\mathbf{x}_{\text{iter}})} \nabla_i F(\mathbf{x}_{\text{new}})$$

   $\mathbf{x}_{\text{new}}(i_2) \leftarrow 1$ 
   $\Delta = F(\mathbf{x}_{\text{new}}) - F(\mathbf{x}_{\text{iter}})$ 
  if  $\Delta \leq 0$  then
     $\mathbf{x}_{\text{iter}+1} \leftarrow \mathbf{x}_{\text{new}}$ 
  else
    if  $\exp(-\Delta/T) > \text{rand}(1)$  then
       $\mathbf{x}_{\text{iter}+1} \leftarrow \mathbf{x}_{\text{new}}$ 
    else
       $\mathbf{x}_{\text{iter}+1} \leftarrow \mathbf{x}_{\text{iter}}$ 
       $\mathbf{x}_{\text{new}} \leftarrow \mathbf{x}_{\text{iter}}$ 
    end if
  end if
end while
return  $\mathbf{x}_{\text{opt}} = \arg \min_{0 \leq \text{iter} \leq \max \text{Iter}} F(\mathbf{x}_{\text{iter}})$ 

```

ALGORITHM 2: SASPAR.

```

Require:  $\max \text{Iter}, \mathbf{x}_0$ 
Initialization:  $\mathbf{x}_{\text{new}} = \mathbf{x}_0$ 
for  $\text{iter} = 0$  to  $\max \text{Iter}$  do
  Select  $i_1$  with positive maximal gradient.
    
$$i_1 = \arg \max_{i \in S(\mathbf{x}_{\text{iter}})} \nabla_i F(\mathbf{x}_{\text{iter}})$$

   $\mathbf{x}_{\text{new}}(i_1) \leftarrow 0$ 
  Select  $i_2$  such that
    
$$i_2 = \arg \min_{i \in S^c(\mathbf{x}_{\text{iter}})} \nabla_i F(\mathbf{x}_{\text{new}})$$

   $\mathbf{x}_{\text{new}}(i_2) \leftarrow 1$ 
   $\Delta = F(\mathbf{x}_{\text{new}}) - F(\mathbf{x}_{\text{iter}})$ 
  if  $\Delta \leq 0$  then
     $\mathbf{x}_{\text{iter}+1} = \mathbf{x}_{\text{new}}$ 
  else
    Terminate the algorithm or restart.
  end if
end for
return  $\mathbf{x}_{\text{opt}} = \arg \min_{0 \leq \text{iter} \leq \max \text{Iter}} F(\mathbf{x}_{\text{iter}})$ 

```

ALGORITHM 3: Greedy 2-opt.

algorithms specified for binary phase retrieval. Since we lack specified existing algorithms, the state-of-the-art algorithm for sparse phase retrieval GESPAR is employed. SASPAR is compared with GESPAR by embedding binary signals to the field of real numbers; thus combinatorial optimization is converted to continuous optimization.

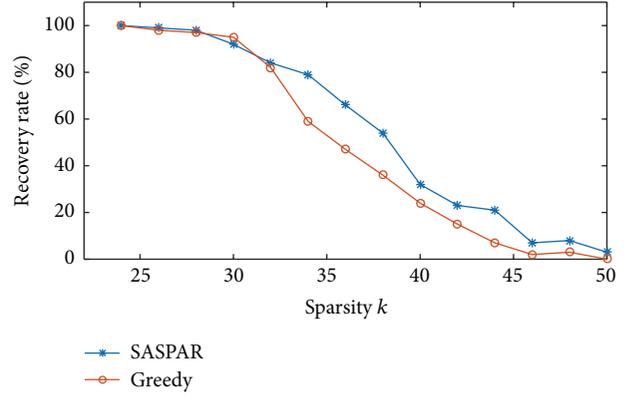


FIGURE 1: SASPAR versus greedy 2-opt.

Fourier measurements are employed in the section. The initial temperature of SASPAR is usually set to be half of the signal length. The stopping temperature and cooling parameter are set to $T_{\text{end}} = 20$ and $\alpha = 0.99$ constantly. Time for iterations of all algorithms in this section is limited to 20 seconds by default. After the cooling procedure is terminated, the SASPAR will restart until it exceeds the time limit.

The evaluation includes its comparison to Algorithm 3 (greedy 2-opt), the sensitivity to max time allowed, recovery from fewer measurements (compared with GESPAR), and robustness to noise (compared with GESPAR). The following experiments run on desktops with i7 4790 CPU and 4 GB RAM. All the scripts implemented by MATLAB for the experiments are available online <https://github.com/dynamex0098/binary-sparse-phase-retrieval-SA.git>.

4.1. Deterministic versus Heuristic. The deterministic greedy algorithm (greedy 2-opt) and the heuristic simulated annealing algorithm (SASPAR) are compared. The max time allowed is fixed to 20 seconds for both algorithms.

Figure 1 illustrates that the greedy 2-opt algorithm is still a competitive algorithm to SASPAR. Their performances are quite close to each other. The success rate is high with sparsity from 24 to 30. However, SASPAR outperforms the 2-opt algorithm from sparsity $k = 32$, which might be explained by the fact that SASPAR requires less time to obtain the minimum while 2-opt algorithm takes more.

4.2. Max Time Allowed for Computation. As is known, more time allowed for searching the optimum results in a higher success rate. In this experiment, signals are recovered from magnitudes of Fourier measurements. The sparsity is fixed to 30 and we double the amount of sampling by adding zeros to signals, which also double the length of signals. The size of signals is set to 256, 512, 1280, and 2048 separately. The max time allowed varies from 2 seconds to 20 seconds. For each length and max time allowed, the computation is repeated 100 times. The result is displayed in Figure 2.

Although the recovery curve in Figure 2 is not smooth, it is clear that a higher recovery rate is obtained as the time allowed increases. Sparser signals tend to be recovered more successfully. 20 seconds is enough to guarantee a high success

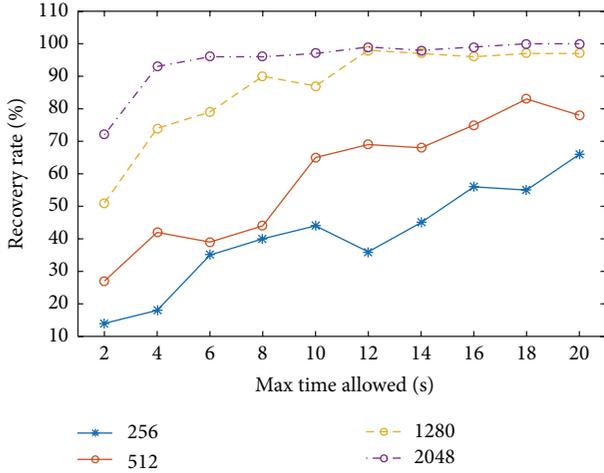


FIGURE 2: Max time allowed versus recovery rate.

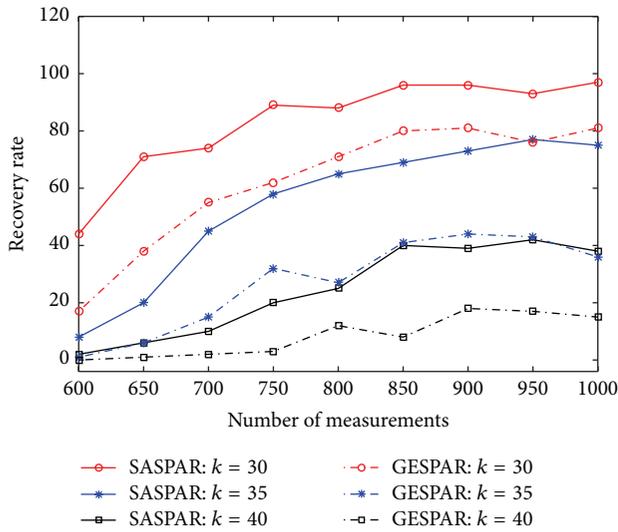


FIGURE 3: Recover from fewer Fourier measurements: SASPAR versus GESPAR.

rate of recovering a 2048-length signal with 30 nonzeros while a 256-length signal with 30 nonzeros might cost more time to guarantee an acceptable success rate.

4.3. Fewer Fourier Measurements. The high frequency part of Fourier transform might not be available due to the limit of devices or other restrictions. It becomes more difficult to reconstruct signals from fewer Fourier measurements. A comparison between SASPAR and GESPAR is conducted. The length of support of signals is fixed to 512 while the measurements vary from 600 to 1000 (by concatenate corresponding zeros to the support of original signals). The sparsity is set to 30, 35, and 40 separately. The experiment is repeated 100 times. Figure 3 illustrates that SASPAR outperforms GESPAR in the given conditions. SASPAR could also maintain a higher recovery rate with fewer measurements.

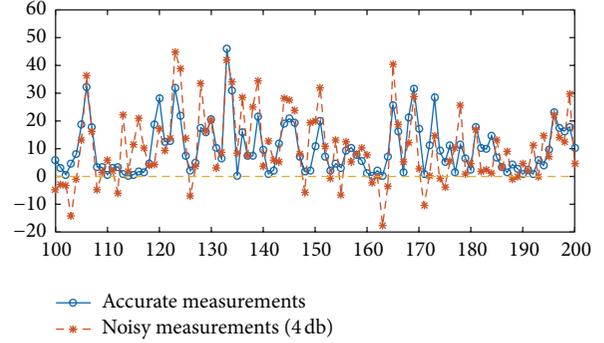


FIGURE 4: An illustration of noisy measurements.

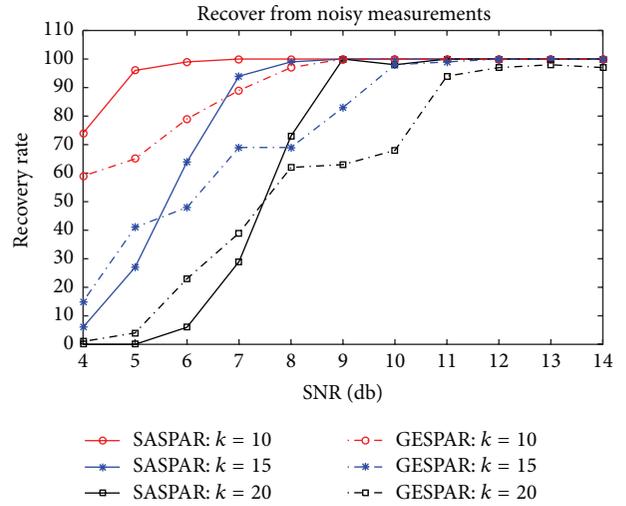


FIGURE 5: SNR versus recovery rate.

4.4. Recover from Noisy Measurements. SASPAR and GESPAR are compared to recover noisy signals with 4 db to 14 db SNR. The length of signals is fixed to 256 and 256 zeros are concatenated for Fourier oversampling. The experiment is repeated 100 times. The Gaussian white noise is added to measurement y . Figure 4 displays an illustration of a noisy signal with 4 db SNR.

To our surprise, the binary phase retrieval is quite robust to noises. As Figure 5 shows, we could recover signals; even the SNR is as low as 4 db, where the signal seems totally overwhelmed by the noise. With proper algorithms selected, sparse binary signals are able to be recovered from seriously noisy signals. However, we still lack the theoretical explanations for robustness of binary sparse phase retrieval.

5. Conclusion

In this paper, we introduce the binary sparse phase retrieval model and present the efficient heuristic algorithm SASPAR. We specified the settings of the simulated annealing procedure and conducted a series of numerical experiments in various cases to testify its efficiency, robustness, and scalability.

Though SASPAR is employed to solve sparse binary phase retrieval problems, it should not be restricted to the binary case, which is quite special in phase retrieval. It might provide an efficient support selection strategy for all sparse recovery problems. As is accepted widely, GESPAR works significantly well in small scale problems. However, we notice that GESPAR requires more measurements for recovery than SASPAR. It implies that the greedy 2-opt support selection strategy might hinder its performance as experiment 1 suggests. Therefore, SASPAR will be adapted to recover the sparse real or complex cases in the future work. In addition, the sparse binary model is also attractive for its simplicity and robustness. As little theoretical evidence has been provided, our future work will also focus on the analysis of binary phase retrieval.

Appendix

A. Estimation of the Sparsity k

Suppose that complex matrix $G \in \mathbb{C}^{m \times n}$ and $G = A + iB$, where $A, B \in \mathbb{R}^{m \times n}$. A, B are composed of *i.i.d.* standard real Gaussian variables $\mathcal{N}(0, 1)$. Let $\|\mathbf{x}\|_0 = k$. Denote $G = [\mathbf{g}_1, \mathbf{g}_2, \dots, \mathbf{g}_m]^H$. Then

$$\begin{aligned} \sum_{i=1}^m y_i &= \sum_{i=1}^m |\mathbf{g}_i^H \mathbf{x}|^2 = \sum_{i=1}^m \left(\sum_{j=1}^n (a_j x_j)^2 + \sum_{j=1}^n (b_j x_j)^2 \right) \\ &\sim 2km\mathbf{U}^2, \end{aligned} \quad (\text{A.1})$$

where random variable $\mathbf{U} \sim \mathcal{N}(0, 1)$, since

$$k = \mathbb{E} \left[\frac{\sum_{i=1}^m y_i}{2m\mathbf{U}^2} \right] \approx \frac{\sum_{i=1}^m y_i}{2m}. \quad (\text{A.2})$$

B. Calculation of the Gradient $\nabla F(\mathbf{x})$

Define the objective function

$$F(\mathbf{x}) = \sum_{i=1}^m 2 \left(|\mathbf{a}_i^H \mathbf{x}|^2 - y_i \right)^2 = \sum_{i=1}^m 4 \left(\mathbf{x}^T \mathbf{a}_i \mathbf{a}_i^H \mathbf{x} - y_i \right)^2. \quad (\text{B.1})$$

Suppose that $\mathbf{a}_i = \mathbf{b}_i + j\mathbf{c}_i$, where \mathbf{b}_i and \mathbf{c}_i are real vectors, $j = \sqrt{-1}$. Then the gradient is computed via the chain rule,

$$\begin{aligned} \nabla F(\mathbf{x}) &= \sum_{i=1}^m 2 \left(\mathbf{x}^T \mathbf{a}_i \mathbf{a}_i^H \mathbf{x} - y_i \right) \cdot \left(2\mathbf{b}_i \mathbf{b}_i^T \mathbf{x} + 2\mathbf{c}_i \mathbf{c}_i^T \mathbf{x} \right) \\ &= 4 \sum_{i=1}^m \left[\mathbf{b}_i \left(|\mathbf{a}_i^H \mathbf{x}|^2 - y_i \right) \cdot \mathbf{b}_i^T \mathbf{x} + \mathbf{c}_i \left(|\mathbf{a}_i^H \mathbf{x}|^2 - y_i \right) \right. \\ &\quad \left. \cdot \mathbf{c}_i^T \mathbf{x} \right] = 4\mathbf{B}^T \left((|\mathbf{A}\mathbf{x}| \odot |\mathbf{A}\mathbf{x}| - \mathbf{y}) \odot \mathbf{B}\mathbf{x} \right) \\ &\quad + 4\mathbf{C}^T \left((|\mathbf{A}\mathbf{x}| \odot |\mathbf{A}\mathbf{x}| - \mathbf{y}) \odot \mathbf{C}\mathbf{x} \right). \end{aligned} \quad (\text{B.2})$$

Competing Interests

The authors declare that they have no competing interests.

Acknowledgments

The authors are grateful for the support from National Natural Science Foundation of China (nos. 61571008 and 61201327).

References

- [1] R. W. Harrison, "Phase problem in crystallography," *Journal of the Optical Society of America A*, vol. 10, no. 5, pp. 1046–1055, 1993.
- [2] A. Walther, "The question of phase retrieval in optics," *Journal of Modern Optics*, vol. 10, no. 1, pp. 41–49, 1963.
- [3] R. Lawrence and B.-H. Juang, *Fundamentals of Speech Recognition*, 1993.
- [4] C. Fienup and J. Dainty, "Phase retrieval and image reconstruction for astronomy," in *Image Recovery: Theory and Application*, pp. 231–275, Academic Press, New York, NY, USA, 1987.
- [5] R. W. Gerchberg and W. O. Saxton, "A practical algorithm for the determination of the phase from image and diffraction plane pictures," *Optik*, vol. 35, no. 2, pp. 237–250, 1972.
- [6] J. R. Fienup, "Phase retrieval algorithms: a comparison," *Applied Optics*, vol. 21, no. 15, pp. 2758–2769, 1982.
- [7] H. H. Bauschke, P. L. Combettes, and D. R. Luke, "Hybrid projection-reflection method for phase retrieval," *Journal of the Optical Society of America A: Optics and Image Science, and Vision*, vol. 20, no. 6, pp. 1025–1034, 2003.
- [8] H. H. Bauschke, P. L. Combettes, and D. R. Luke, "Phase retrieval, error reduction algorithm, and Fienup variants: a view from convex optimization," *Journal of the Optical Society of America A*, vol. 19, no. 7, pp. 1334–1345, 2002.
- [9] E. J. Candes, Y. Eldar, T. Strohmer, and V. Voroninski, "Phase retrieval via matrix completion," *SIAM Review*, vol. 57, no. 2, pp. 225–251, 2015.
- [10] E. J. Candès, X. Li, and M. Soltanolkotabi, "Phase retrieval via wirtinger flow: theory and algorithms," *IEEE Transactions on Information Theory*, vol. 61, no. 4, pp. 1985–2007, 2015.
- [11] P. Netrapalli, P. Jain, and S. Sanghavi, "Phase retrieval using alternating minimization," in *Proceedings of the 27th Annual Conference on Neural Information Processing Systems (NIPS '13)*, December 2013.
- [12] P. Schniter and S. Rangan, "Compressive phase retrieval via generalized approximate message passing," *IEEE Transactions on Signal Processing*, vol. 63, no. 4, pp. 1043–1055, 2015.
- [13] Y. Shechtman, A. Beck, and Y. C. Eldar, "GESPAR: efficient phase retrieval of sparse signals," *IEEE Transactions on Signal Processing*, vol. 62, no. 4, pp. 928–938, 2014.
- [14] S. Mukherjee and C. S. Seelamantula, "An iterative algorithm for phase retrieval with sparsity constraints: application to frequency domain optical coherence tomography," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '12)*, pp. 553–556, Kyoto, Japan, March 2012.
- [15] Y. Shechtman, Y. C. Eldar, A. Szameit, and M. Segev, "Sparsity based sub-wavelength imaging with partially incoherent light via quadratic compressed sensing," *Optics Express*, vol. 19, no. 16, pp. 14807–14822, 2011.
- [16] H. Ohlsson and Y. C. Eldar, "On conditions for uniqueness in sparse phase retrieval," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '14)*, pp. 1841–1845, IEEE, Florence, Italy, May 2014.

- [17] Y. C. Eldar, P. Sidorenko, D. G. Mixon, S. Barel, and O. Cohen, "Sparse phase retrieval from short-time fourier measurements," *IEEE Signal Processing Letters*, vol. 22, no. 5, pp. 638–642, 2015.
- [18] U. Nakarmi and N. Rahnavard, "BCS: compressive sensing for binary sparse signals," in *Proceedings of the IEEE Military Communications Conference (MILCOM '12)*, pp. 1–5, Orlando, Fla, USA, November 2012.
- [19] D. Pham and D. Karaboga, *Intelligent Optimisation Techniques: Genetic Algorithms, Tabu Search, Simulated Annealing and Neural Networks*, Springer Science & Business Media, 2012.
- [20] N. Yoshikawa and T. Yatagai, "Phase optimization of a kinoform by simulated annealing," *Applied Optics*, vol. 33, no. 5, pp. 863–868, 1994.
- [21] D. Morris, "Simulated annealing applied to the Misell algorithm for phase retrieval," *IEE Proceedings—Microwaves, Antennas and Propagation*, vol. 143, no. 4, pp. 298–303, 1996.
- [22] J. Turunen, A. Vasara, and J. Westerholm, "Kinoform phase relief synthesis: a stochastic method," *Optical Engineering*, vol. 28, no. 11, Article ID 281162, 1989.
- [23] M. Nieto-Vesperinas, F. J. Fuentes, and R. Navarro, "Performance of a simulated-annealing algorithm for phase retrieval," *JOSA A*, vol. 5, no. 1, pp. 30–38, 1988.
- [24] Y. Lin, T. J. Kessler, and G. N. Lawrence, "Design of continuous surface-relief phase plates by surface-based simulated annealing to achieve control of focal-plane irradiance," *Optics Letters*, vol. 21, no. 20, pp. 1703–1705, 1996.
- [25] Y. Nourani and B. Andresen, "A comparison of simulated annealing cooling strategies," *Journal of Physics A: Mathematical and General*, vol. 31, no. 41, pp. 8373–8385, 1998.
- [26] C. H. Papadimitriou and K. Steiglitz, *Combinatorial Optimization: Algorithms and Complexity*, Courier Corporation, 1982.




Hindawi

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

