

PRIME: Phase Retrieval via Majorization-Minimization

Tianyu Qiu, Prabhu Babu, and Daniel P. Palomar, *Fellow, IEEE*

Abstract—This paper considers the phase retrieval problem in which measurements consist of only the magnitude of several linear measurements of the unknown, e.g., spectral components of a time sequence. We develop low-complexity algorithms with superior performance based on the majorization-minimization (MM) framework. The proposed algorithms are referred to as PRIME: Phase Retrieval via the Majorization-minimization technique. They are preferred to existing benchmark methods since at each iteration a simple surrogate problem is solved with a closed-form solution that monotonically decreases the original objective function. In total, three algorithms are proposed using different majorization-minimization techniques. Experimental results validate that our algorithms outperform existing methods in terms of successful recovery and mean-square error under various settings.

Index Terms—Phase retrieval, majorization-minimization, convex optimization.

I. INTRODUCTION

PHASE retrieval, the recovery of a signal from the magnitude of linear measurements like its Fourier transform, arises in various applications such as optical imaging [1], crystallography [2], microscopy [3], and audio signal processing [4]–[6]. In general, optical devices (e.g., CCD cameras, human eyes, etc.) can record the intensity of the incoming light but not the phase, hence it is challenging to uniquely recover the original signal without phase information.

Mathematically speaking, the phase retrieval problem is to recover a K -dimensional complex signal $\mathbf{x} \in \mathbb{C}^K$ from the magnitude of N linear measurements (usually corrupted with noise):

$$y_i = |\mathbf{a}_i^H \mathbf{x}|^2 + n_i \in \mathbb{R}, \quad i = 1, \dots, N, \quad (1)$$

where the measurement vectors $\{\mathbf{a}_i \in \mathbb{C}^K\}_{i=1}^N$ are known beforehand. In the Fourier transform case, they correspond to rows of the Discrete Fourier Transform (DFT) matrix. In a more general case, they can be any vectors of interest. Due to the loss of phase information, the number of measurements should exceed the dimension of the original signal in order to successfully recover the signal. The authors of [7] proved that the number of

measurements N should at least be on the order of $K \log K$ for a successful recovery with high probability when the measurement vectors are chosen independently and uniformly at random on the unit sphere. A conjecture is posed in [8] that $4K - 4$ measurements are necessary and sufficient for injectivity, i.e., to uniquely recover the original signal (up to a constant phase shift) when provided with multiple measurements.

Numerical methods to recover the original signal \mathbf{x} from multiple measurements $\{y_i\}_{i=1}^N$ fall mainly into two categories. The first is based on the Gerchberg-Saxton algorithm [9]–[12], and solves the phase retrieval problem through alternating minimizations. The second and more recent class is based on semidefinite relaxation [7], [13], [14]. The idea is to recover the original signal through a convex semidefinite programming (SDP) problem by introducing a rank-1 matrix $\mathbf{X} := \mathbf{x}\mathbf{x}^H$, named “matrix-lifting.” Unfortunately, the increase of dimension in this matrix-lifting procedure limits the application of the algorithm to small scale problems and it is not appropriate for big data problems. A novel work [12] combined the structure of the Gerchberg-Saxton algorithm and the idea of SDP relaxation. They lifted the phase vector instead of the original signal to formulate a tractable convex relaxation for the original non-convex quadratic problem, and solved it through a provably convergent block coordinate descent algorithm where each iteration is only a matrix vector product. More recently, [15] proposed to solve the phase retrieval problem using the steepest descent method with a heuristic step size. Interestingly, one of the algorithms we present in this paper turns out to have similar updating rules but with a clearly specified step size. Besides these, other methods further exploit the signal sparsity: [16] combined the damped Gauss-Newton method and “2-opt” method to retrieve the phase of a sparse signal and [17] employed a probabilistic approach based on the generalized approximate message passing.

In this paper, we propose methods to solve the phase retrieval problem using different majorization-minimization (MM) techniques [18]. Instead of dealing with the original cumbersome optimization problem directly, an MM algorithm optimizes a sequence of simple surrogate problems successively. The sequence of points generated by solving each surrogate problem is guaranteed to converge to a stationary point of the original problem. As for the phase retrieval problem, by majorizing certain terms in the objective function, we manage to substitute the original non-convex and difficult problem with different convex optimization problems. All these surrogate problems are designed to share the same favorable property of having a simple closed-form solution and only require basic matrix multiplications at every iteration. Different from the SDP approach, our algorithms do not require matrix-lifting, and at every iteration

Manuscript received May 11, 2015; revised February 17, 2016 and June 10, 2016; accepted June 19, 2016. Date of publication June 24, 2016; date of current version August 11, 2016. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Mark Plumbley. This work was supported by the Hong Kong RGC 16206315 research grant. Preliminary results of this work were published in the Forty-Ninth Asilomar Conference on Signals, Systems, and Computers, November 2015.

T. Qiu and D. P. Palomar are with The Hong Kong University of Science and Technology (HKUST), Hong Kong (e-mail: tqiu@ust.hk; palomar@ust.hk).

P. Babu is with the CARE, IIT Delhi, Delhi India (e-mail: prabhubabu@care.iitd.ac.in).

Color versions of one or more of the figures in this letter are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TSP.2016.2585084

yield a simple closed-form solution directly for the signal \mathbf{x} . Therefore our algorithms can be applied to very large scale problems in big data systems.

The contributions of this paper are:

- 1) Numerical methods for two different objectives of the phase retrieval problem (to recover the original signal from either the modulus squared or modulus of its linear measurements).
- 2) Monotonicity and provable convergence to a stationary point of the sequence of points generated by our algorithms.
- 3) Much faster numerical convergence of our algorithms compared to the Wirtinger Flow algorithm and the Gerchberg-Saxton algorithm.
- 4) Low complexity per iteration of our algorithms (only requiring basic matrix multiplication).

The remaining sections are organized as follows. In Section II, we present two different problem formulations for the phase retrieval problem. In Section III, after a brief overview of the general MM framework is introduced, we propose algorithms for both problems via different majorization-minimization techniques. An acceleration scheme is discussed in Section IV to further increase the convergence speed of our algorithms. Finally, in Section V, we provide the numerical results under various settings, e.g., different measurement matrices, clean measurements and noisy measurements.

Notation: Boldface upper case letters (e.g., \mathbf{X} , \mathbf{A}) denote matrices, while boldface lower case letters (e.g., \mathbf{x} , \mathbf{a}) denote column vectors, and italics (e.g., x , a , D) denote scalars. \mathbb{R} and \mathbb{C} denote the real field and the complex field, respectively. For any complex number x , $|x|$ denotes the magnitude, and $\arg(x)$ denotes the phase. As for vectors, $|\mathbf{x}|$ denotes the element-wise magnitude and $\arg(\mathbf{x})$ denotes the element-wise phase. The superscripts $(\cdot)^T$, (\cdot) and $(\cdot)^H$ denote the transpose, complex conjugate and conjugate transpose, respectively. X_{ij} or $[\mathbf{X}]_{ij}$ denotes the element at the i -th row and j -th column of a matrix \mathbf{X} , and x_i or $[\mathbf{x}]_i$ denotes the i -th element of a vector \mathbf{x} . $\text{Tr}(\cdot)$ is the trace of a matrix. $\text{diag}(\mathbf{x})$ is a diagonal matrix formed by setting vector \mathbf{x} as its principal diagonal, while $\text{diag}(\mathbf{X})$ is a column vector consisting of all the elements in the principal diagonal of matrix \mathbf{X} . The column vector $\text{vec}(\mathbf{X})$ is formed by stacking all the columns of a matrix \mathbf{X} . As usual, the Euclidean norm of a vector \mathbf{x} is denoted by $\|\mathbf{x}\|$. The curled inequality symbol \succeq (its reverse form \preceq) is used to denote generalized inequality; $\mathbf{A} \succeq \mathbf{B}$ ($\mathbf{B} \preceq \mathbf{A}$) means that $\mathbf{A} - \mathbf{B}$ is a Hermitian positive semidefinite matrix. \mathbf{I}_n is the $n \times n$ identity matrix, and $\mathbf{1}$ is a vector with all elements one.

II. PROBLEM FORMULATION AND EXISTING METHODS

As described previously, the phase retrieval problem is to recover a complex signal from magnitudes of its linear measurements. In general, it is difficult to solve the problem due to the missing phase information. In this paper, we consider the case in which we have multiple measurements. Usually these measurements $\{y_i\}_{i=1}^N$ are corrupted with noise. When the noise follows a Gaussian distribution, a general choice is to consider

the following least squares problem, which coincides with the maximum likelihood estimation of the original signal [14], [15]:

$$\underset{\mathbf{x}}{\text{minimize}} \quad f(\mathbf{x}) := \sum_{i=1}^N \left| y_i - |\mathbf{a}_i^H \mathbf{x}|^2 \right|^2. \quad (2)$$

Here, the measurement vectors $\{\mathbf{a}_i \in \mathbb{C}^K\}_{i=1}^N$ are known beforehand and can be any vectors of interest. In this paper, we consider two different cases. The first is the traditional Fourier transform case, in which $\{\mathbf{a}_i^H\}_{i=1}^N$ correspond to rows of the DFT matrix; i.e., the k -th element in vector \mathbf{a}_i is $[\mathbf{a}_i]_k = e^{j2\pi(k-1)(i-1)/N}$. The second is the random matrix case, in which $\{\mathbf{a}_i\}_{i=1}^N$ are regarded as standard complex Gaussian distributed. Specifically, every element in the measurement vectors is a random variable in which both the real and imaginary parts are drawn from the standard Gaussian distribution $\mathcal{N}(0, 1)$ independently.

Defining the measurement matrix

$$\mathbf{A} = [\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_N] \in \mathbb{C}^{K \times N}, \quad (3)$$

and stacking the multiple measurements $\{y_i\}_{i=1}^N$ as a vector \mathbf{y} , we can formulate the phase retrieval problem (2) in a more compact form:

$$\underset{\mathbf{x}}{\text{minimize}} \quad \left\| \mathbf{y} - |\mathbf{A}^H \mathbf{x}|^2 \right\|_2^2. \quad (4)$$

Notice that here the operator $|\cdot|$ is applied element-wise when the argument is a vector (similarly for $(\cdot)^2$).

The authors of [15] proposed the following Wirtinger Flow algorithm based on the gradient descent method to solve problem (2). They chose the leading eigenvector of $\mathbf{A} \text{diag}(\mathbf{y}) \mathbf{A}^H$ as the initial point because it would coincide with the optimal solution provided infinite samples ($N \rightarrow +\infty$) by the strong law of large numbers. In the algorithm, μ is a scheduling term. Instead of using the heuristic value in [15], we adopt the backtracking method to find a suitable value.

Algorithm 1: The Wirtinger Flow Algorithm.

Input: \mathbf{A} , \mathbf{y} , t_0 (maximum iteration number)

1: Initial $\mathbf{x}^{(0)} \leftarrow$ leading eigenvector of $\mathbf{A} \text{diag}(\mathbf{y}) \mathbf{A}^H$

2: Set constant $\lambda^2 \leftarrow K \sum_{i=1}^N y_i / \sum_{i=1}^N \|\mathbf{a}_i\|^2$

3: $\mathbf{x}^{(0)} \leftarrow \lambda \mathbf{x}^{(0)}$

4: **for** $k = 0, \dots, t_0 - 1$ **do**

5: $\nabla f = 4\mathbf{A} \text{diag}(|\mathbf{A}^H \mathbf{x}^{(k)}|^2 - \mathbf{y}) \mathbf{A}^H \mathbf{x}^{(k)}$

6: $\mu = 1$

7: $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} - \mu \nabla f$

8: **while** $f(\mathbf{x}^{(k+1)}) > f(\mathbf{x}^{(k)}) - 0.01\mu \|\nabla f\|_2^2$ **do**

9: $\mu \leftarrow \mu/2$

10: $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} - \mu \nabla f$

11: **end while**

12: **end for**

Output: $\mathbf{x}^{(t_0)}$.

Different from problem (2), an alternative is to solve the following problem using the modulus, as opposed to the squared

modulus, of the linear measurements of the signal [9]–[12]:

$$\underset{\mathbf{x}}{\text{minimize}} \quad \|\sqrt{\mathbf{y}} - |\mathbf{A}^H \mathbf{x}|\|_2^2, \quad (5)$$

where the operator $\sqrt{\cdot}$ is applied element-wise when the argument is a vector. As pointed out in [11], [12], if we had access to the phase information $\mathbf{c} \in \mathbb{C}^N$ of the linear measurements $\mathbf{A}^H \mathbf{x}$ (i.e., $c_i = e^{j \arg(\mathbf{a}_i^H \mathbf{x})}$) and $N \geq K$, then problem (5) would reduce to one of solving a system of linear equations

$$\text{diag}(\sqrt{\mathbf{y}})\mathbf{c} = \mathbf{A}^H \mathbf{x}, \quad (6)$$

Of course we do not know this phase vector \mathbf{c} ; hence one intuitive approach is to solve the following problem by introducing a new variable $\mathbf{c} \in \mathbb{C}^N$ representing the phase information:

$$\begin{aligned} &\underset{\mathbf{x}, \mathbf{c}}{\text{minimize}} \quad \|\mathbf{A}^H \mathbf{x} - \text{diag}(\sqrt{\mathbf{y}})\mathbf{c}\|_2^2 \\ &\text{subject to} \quad |c_i| = 1, \quad i = 1, \dots, N. \end{aligned} \quad (7)$$

Note that the above problem (7) is not convex because the vector \mathbf{c} is restricted to be phases; i.e., all the elements are limited to be of magnitude one. One classical approach is to use the Gerchberg-Saxton algorithm [9], thus alternately updating \mathbf{x} and \mathbf{c} so as to minimize problem (7). For a given \mathbf{c} , problem (7) reduces to a standard least squares problem, which can be solved easily. For a fixed \mathbf{x} , the optimal solution for \mathbf{c} is $\mathbf{c}^* = e^{j \arg(\mathbf{A}^H \mathbf{x})}$. Here both the operators $e^{(\cdot)}$ and $\arg(\cdot)$ are applied element-wise.

Algorithm 2: The Gerchberg-Saxton Algorithm.

Input: \mathbf{A} , \mathbf{y} , t_0

1: Initial $\mathbf{x}^{(0)} \leftarrow$ leading eigenvector of $\mathbf{A} \text{diag}(\mathbf{y}) \mathbf{A}^H$

2: **for** $k = 0, \dots, t_0 - 1$ **do**

3: $\mathbf{c}^{(k+1)} = e^{j \arg(\mathbf{A}^H \mathbf{x}^{(k)})}$

4: $\mathbf{x}^{(k+1)} \leftarrow \arg_{\mathbf{x}} \min \|\mathbf{A}^H \mathbf{x} - \text{diag}(\sqrt{\mathbf{y}})\mathbf{c}^{(k+1)}\|_2^2$

5: **end for**

Output: $\mathbf{x}^{(t_0)}$.

In the next section, we are going to develop three algorithms using different MM techniques for both problems (2) and (5). Experimental results show that our algorithms outperform the benchmark algorithms (the Wirtinger Flow algorithm and the Gerchberg-Saxton algorithm) in terms of successful recovery probability and mean squared error.

III. PHASE RETRIEVAL VIA MAJORIZATION-MINIMIZATION

In this section, we first provide a concise introduction on the general MM framework, after which we present our algorithms for problems (2) and (5). In total, three different algorithms are proposed, one for problem (2) and two for problem (5). For problem (5), one of the algorithms turns out to be exactly the same as the Gerchberg-Saxton algorithm. The other algorithm turns out to have similar updating rules to those of the Wirtinger Flow algorithm, but unlike in [15], where a heuristic step size is used, our algorithm has a clearly specified step size. For problem (2), our algorithm formulates it as the leading eigenvector problem.

A. The MM Algorithm

The majorization-minimization (MM) algorithm [18] is a generalization of the well-known expectation-maximization (EM) algorithm. Instead of dealing with the original difficult optimization problem directly, an MM algorithm solves a series of simple surrogate optimization problems, producing a series of points that drive the original objective function downhill.

For a real valued function $f(\boldsymbol{\theta})$, any function $g(\boldsymbol{\theta} | \boldsymbol{\theta}^{(m)})$ that satisfies the following two conditions is said to be a majorization function of $f(\boldsymbol{\theta})$ at the point $\boldsymbol{\theta}^{(m)}$:

$$\begin{aligned} g(\boldsymbol{\theta} | \boldsymbol{\theta}^{(m)}) &\geq f(\boldsymbol{\theta}) \text{ for all } \boldsymbol{\theta}, \\ g(\boldsymbol{\theta}^{(m)} | \boldsymbol{\theta}^{(m)}) &= f(\boldsymbol{\theta}^{(m)}). \end{aligned} \quad (8)$$

That is to say, the function $g(\boldsymbol{\theta} | \boldsymbol{\theta}^{(m)})$ is a global upper bound of the function $f(\boldsymbol{\theta})$, and touches it at the point $\boldsymbol{\theta}^{(m)}$. Instead of dealing with the original function $f(\boldsymbol{\theta})$ directly, which is usually non-convex or non-differentiable, an MM algorithm optimizes the sequence of majorization functions $\{g(\boldsymbol{\theta} | \boldsymbol{\theta}^{(m)})\}$. In general, these majorization functions are chosen to be convex and differentiable and much easier to solve, e.g., yielding a simple closed-form solution. Initialized by any feasible point $\boldsymbol{\theta}^{(0)}$, a sequence of points $\{\boldsymbol{\theta}^{(m)}\}$ is generated by the MM algorithm following the update rule:

$$\boldsymbol{\theta}^{(m+1)} \in \arg \min_{\boldsymbol{\theta}} g(\boldsymbol{\theta} | \boldsymbol{\theta}^{(m)}). \quad (9)$$

A favorable property of the MM algorithm is that the sequence of points $\{\boldsymbol{\theta}^{(m)}\}$ generated by minimizing the majorization functions $\{g(\boldsymbol{\theta} | \boldsymbol{\theta}^{(m)})\}$ drive $f(\boldsymbol{\theta})$ downhill:

$$\begin{aligned} f(\boldsymbol{\theta}^{(m+1)}) &\leq g(\boldsymbol{\theta}^{(m+1)} | \boldsymbol{\theta}^{(m)}) \leq g(\boldsymbol{\theta}^{(m)} | \boldsymbol{\theta}^{(m)}) \\ &= f(\boldsymbol{\theta}^{(m)}). \end{aligned} \quad (10)$$

The first inequality and the third equality are a direct application of the definition of the majorization function in (8). The second inequality comes from (9) that $\boldsymbol{\theta}^{(m+1)}$ is a minimizer of $g(\boldsymbol{\theta} | \boldsymbol{\theta}^{(m)})$. Hence under the MM framework, one can find a stationary point for the original function by solving the majorization functions instead.

B. PRIME-Modulus-Single-Term

We first apply the MM techniques to problem (5). By expanding the objective function

$$\|\sqrt{\mathbf{y}} - |\mathbf{A}^H \mathbf{x}|\|_2^2 = \sum_{i=1}^N \left(|\mathbf{a}_i^H \mathbf{x}|^2 - 2\sqrt{y_i} |\mathbf{a}_i^H \mathbf{x}| + y_i \right) \quad (11)$$

and discarding the constant term $\sum_{i=1}^N y_i$, problem (5) is equivalent to

$$\underset{\mathbf{x}}{\text{minimize}} \quad \sum_{i=1}^N \left(|\mathbf{a}_i^H \mathbf{x}|^2 - 2\sqrt{y_i} |\mathbf{a}_i^H \mathbf{x}| \right). \quad (12)$$

Here we keep the first term $\sum_{i=1}^N |\mathbf{a}_i^H \mathbf{x}|^2$ and only majorize the (nonconvex) second term $-\sum_{i=1}^N 2\sqrt{y_i} |\mathbf{a}_i^H \mathbf{x}|$. According

to the Cauchy-Schwarz inequality

$$|\mathbf{a}_i^H \mathbf{x}| \cdot |\mathbf{a}_i^H \mathbf{x}^{(k)}| \geq \operatorname{Re} \left(\mathbf{a}_i^H \mathbf{x} \cdot (\mathbf{x}^{(k)})^H \mathbf{a}_i \right), \quad (13)$$

the second term $-\sum_{i=1}^N 2\sqrt{y_i} |\mathbf{a}_i^H \mathbf{x}|$ can be majorized as

$$-\sum_{i=1}^N 2\sqrt{y_i} |\mathbf{a}_i^H \mathbf{x}| \leq -\sum_{i=1}^N 2\sqrt{y_i} \frac{\operatorname{Re} \left(\mathbf{a}_i^H \mathbf{x} \cdot (\mathbf{x}^{(k)})^H \mathbf{a}_i \right)}{|\mathbf{a}_i^H \mathbf{x}^{(k)}|}. \quad (14)$$

Thus the convex majorization problem for (12) is

$$\underset{\mathbf{x}}{\text{minimize}} \sum_{i=1}^N \left(|\mathbf{a}_i^H \mathbf{x}|^2 - 2\sqrt{y_i} \frac{\operatorname{Re} \left(\mathbf{a}_i^H \mathbf{x} \cdot (\mathbf{x}^{(k)})^H \mathbf{a}_i \right)}{|\mathbf{a}_i^H \mathbf{x}^{(k)}|} \right), \quad (15)$$

which is equivalent to

$$\underset{\mathbf{x}}{\text{minimize}} \sum_{i=1}^N |\mathbf{a}_i^H \mathbf{x} - c_i \sqrt{y_i}|^2, \quad (16)$$

where

$$c_i := \frac{\mathbf{a}_i^H \mathbf{x}^{(k)}}{|\mathbf{a}_i^H \mathbf{x}^{(k)}|} = e^{j \arg(\mathbf{a}_i^H \mathbf{x}^{(k)})}. \quad (17)$$

Note that c_i actually stands for the phase information of the linear measurement $\mathbf{a}_i^H \mathbf{x}^{(k)}$. Therefore, we introduce here the vector $\mathbf{c} = e^{j \arg(\mathbf{A}^H \mathbf{x}^{(k)})}$ and further can formulate problem (16) in the following more compact form:

$$\underset{\mathbf{x}}{\text{minimize}} \|\mathbf{A}^H \mathbf{x} - \operatorname{diag}(\sqrt{\mathbf{y}})\mathbf{c}\|_2^2, \quad (18)$$

which is a simple least squares problem. And it has a simple closed-form solution $\mathbf{x}^* = (\mathbf{A}\mathbf{A}^H)^{-1} \mathbf{A} \operatorname{diag}(\sqrt{\mathbf{y}})\mathbf{c}$ if the measurement matrix \mathbf{A} has full row rank. It is quite interesting that we have managed to solve problem (5) deriving the same Gerchberg-Saxton algorithm but from a totally different majorization-minimization perspective.

C. PRIME-Modulus-Both-Terms

We now consider majorizing both terms in problem (12). In principle, this is not necessary since after majorizing the second term the problem becomes convex with a simple closed-form solution. Also, in general, the fewer terms we majorize, the better it tends to be in terms of convergence. Nevertheless, we explore this option for the sake of an even simpler algorithm, albeit with a potentially slower convergence.

Claim 1: Let \mathbf{L} be a $K \times K$ Hermitian matrix and \mathbf{M} be another $K \times K$ Hermitian matrix such that $\mathbf{M} \succeq \mathbf{L}$. Then for any point $\mathbf{x}_0 \in \mathbb{C}^K$, the quadratic function $\mathbf{x}^H \mathbf{L} \mathbf{x}$ is majorized by $\mathbf{x}^H \mathbf{M} \mathbf{x} + 2\operatorname{Re}(\mathbf{x}^H (\mathbf{L} - \mathbf{M}) \mathbf{x}_0) + \mathbf{x}_0^H (\mathbf{M} - \mathbf{L}) \mathbf{x}_0$ at \mathbf{x}_0 .

Proof: The result holds by simply rearranging the terms in $(\mathbf{x} - \mathbf{x}_0)^H (\mathbf{M} - \mathbf{L})(\mathbf{x} - \mathbf{x}_0) \geq 0$, cf. [19]. ■

The above claim provides a method to majorize the first term $\sum_{i=1}^N |\mathbf{a}_i^H \mathbf{x}|^2$.

$$\begin{aligned} \sum_{i=1}^N |\mathbf{a}_i^H \mathbf{x}|^2 &= \sum_{i=1}^N \mathbf{x}^H \mathbf{a}_i \mathbf{a}_i^H \mathbf{x} = \mathbf{x}^H \mathbf{A} \mathbf{A}^H \mathbf{x} \\ &\leq \lambda_{\max}(\mathbf{A} \mathbf{A}^H) \mathbf{x}^H \mathbf{x} \\ &\quad + 2\operatorname{Re} \left[\mathbf{x}^H (\mathbf{A} \mathbf{A}^H - \lambda_{\max}(\mathbf{A} \mathbf{A}^H) \mathbf{I}) \mathbf{x}^{(k)} \right] \\ &\quad + (\mathbf{x}^{(k)})^H (\lambda_{\max}(\mathbf{A} \mathbf{A}^H) \mathbf{I} - \mathbf{A} \mathbf{A}^H) \mathbf{x}^{(k)}. \end{aligned} \quad (19)$$

Discarding the constant term, the new majorization problem for (12) is

$$\begin{aligned} \underset{\mathbf{x}}{\text{minimize}} \quad &\lambda_{\max}(\mathbf{A} \mathbf{A}^H) \mathbf{x}^H \mathbf{x} + 2\operatorname{Re} \\ &\times \left[\mathbf{x}^H \left(\mathbf{A} \mathbf{A}^H - \lambda_{\max}(\mathbf{A} \mathbf{A}^H) \mathbf{I} - \sum_{i=1}^N \frac{\sqrt{y_i} \mathbf{a}_i \mathbf{a}_i^H}{|\mathbf{a}_i^H \mathbf{x}^{(k)}|} \right) \mathbf{x}^{(k)} \right], \end{aligned} \quad (20)$$

which is equivalent to the following problem:

$$\underset{\mathbf{x}}{\text{minimize}} \|\mathbf{x} - \mathbf{b}\|_2^2, \quad (21)$$

and has a closed-form solution $\mathbf{x}^* = \mathbf{b}$, where the constant

$$\mathbf{b} := \mathbf{x}^{(k)} + \lambda_{\max}^{-1}(\mathbf{A} \mathbf{A}^H) \mathbf{A} \operatorname{diag}(\mathbf{p}) \mathbf{A}^H \mathbf{x}^{(k)}, \quad (22)$$

and $p_i = \sqrt{y_i}/|\mathbf{a}_i^H \mathbf{x}^{(k)}| - 1$. This algorithm is similar to the steepest descent method proposed recently in [15], where the authors chose a heuristic step size. Now we can see that one suitable step size is $\lambda_{\max}^{-1}(\mathbf{A} \mathbf{A}^H)$.

So far there is no preference between these two majorization problems for problem (5). They both yield a simple closed-form solution at every iteration and are preferable under different problem settings. The procedure to solve the first majorization problem (18) turns out to be the same as in the Gerchberg-Saxton algorithm, in which at every iteration one only needs to solve a standard least squares problem. As for the second majorization problem (21), one needs to calculate the leading eigenvalue $\lambda_{\max}(\mathbf{A} \mathbf{A}^H)$, which is cumbersome when the signal to be recovered is of a very large dimension. Fortunately, when the measurement matrix \mathbf{A} is from the DFT matrix, this largest eigenvalue is as simple as $\lambda_{\max}(\mathbf{A} \mathbf{A}^H) = N$ (see Appendix A for the proof). Therefore, PRIME-Modulus-Both-Terms may outperform PRIME-Modulus-Single-Term in the Fourier transform case.

D. PRIME-Power

Now we are going to show step by step how we majorize problem (2) as a leading eigenvector problem using MM techniques. By introducing two matrices $\mathbf{A}_i = \mathbf{a}_i \mathbf{a}_i^H$ and $\mathbf{X} = \mathbf{x} \mathbf{x}^H$, we can rewrite problem (2) as

$$\begin{aligned} \underset{\mathbf{x}, \mathbf{X}}{\text{minimize}} \quad &\sum_{i=1}^N (y_i - \operatorname{Tr}(\mathbf{A}_i \mathbf{X}))^2 \\ \text{subject to} \quad &\mathbf{X} = \mathbf{x} \mathbf{x}^H, \end{aligned} \quad (23)$$

which is equivalent to

$$\begin{aligned} & \underset{\mathbf{X}, \mathbf{X}}{\text{minimize}} \quad \sum_{i=1}^N \left[(\text{Tr}(\mathbf{A}_i \mathbf{X}))^2 - 2y_i \text{Tr}(\mathbf{A}_i \mathbf{X}) \right] \\ & \text{subject to} \quad \mathbf{X} = \mathbf{x}\mathbf{x}^H, \end{aligned} \quad (24)$$

by ignoring the constant term $\sum_{i=1}^N y_i^2$. We choose to majorize the first term $\sum_{i=1}^N (\text{Tr}(\mathbf{A}_i \mathbf{X}))^2$ (note that this term is already convex in \mathbf{X} but the majorization will help in producing a much simpler problem), and keep the second term $\sum_{i=1}^N (-2y_i \text{Tr}(\mathbf{A}_i \mathbf{X}))$ since it is linear in \mathbf{X} .

Note that both matrices \mathbf{X} and \mathbf{A}_i are Hermitian. Thus we can write the first term $\sum_{i=1}^N (\text{Tr}(\mathbf{A}_i \mathbf{X}))^2$ in problem (24) as

$$\begin{aligned} \sum_{i=1}^N (\text{Tr}(\mathbf{A}_i \mathbf{X}))^2 &= \sum_{i=1}^N \text{vec}(\mathbf{X})^H \text{vec}(\mathbf{A}_i) \text{vec}(\mathbf{A}_i)^H \text{vec}(\mathbf{X}) \\ &= \text{vec}(\mathbf{X})^H \Phi \text{vec}(\mathbf{X}), \end{aligned} \quad (25)$$

where we define the matrix

$$\Phi := \sum_{i=1}^N \text{vec}(\mathbf{A}_i) \text{vec}(\mathbf{A}_i)^H. \quad (26)$$

This matrix Φ is just a constant with regard to the variables \mathbf{x} and \mathbf{X} since all the measurement vectors $\{\mathbf{a}_i\}_{i=1}^N$ are known beforehand. According to Claim 1, by treating the matrix Φ as \mathbf{L} and setting $\mathbf{M} = D\mathbf{I}_{K^2}$, where $D \geq \lambda_{\max}(\Phi)$ guarantees $\mathbf{M} \succeq \mathbf{L}$, the expression $\text{vec}(\mathbf{X})^H \Phi \text{vec}(\mathbf{X})$ in (25) can be majorized by the following function (from now on and when no misunderstanding is caused, the dimension in the identity matrix will be omitted for the sake of notation):

$$\begin{aligned} u_1(\mathbf{X} | \mathbf{X}^{(k)}) &= D \text{vec}(\mathbf{X})^H \text{vec}(\mathbf{X}) \\ &\quad + 2\text{Re} \left[\text{vec}(\mathbf{X})^H (\Phi - D\mathbf{I}) \text{vec}(\mathbf{x}^{(k)}) \right] \\ &\quad + \text{vec}(\mathbf{x}^{(k)})^H (D\mathbf{I} - \Phi) \text{vec}(\mathbf{x}^{(k)}) \\ &= D\text{Tr}(\mathbf{X}\mathbf{X}) + 2 \sum_{i=1}^N \text{Tr}(\mathbf{X}\mathbf{A}_i) \text{Tr}(\mathbf{X}^{(k)} \mathbf{A}_i) \\ &\quad - 2D\text{Tr}(\mathbf{X}\mathbf{X}^{(k)}) + \text{const.}, \end{aligned} \quad (27)$$

where *const.* represents a constant term not dependent on \mathbf{X} . Combining this majorization function $u_1(\mathbf{X} | \mathbf{X}^{(k)})$ and the second term $\sum_{i=1}^N (-2y_i \text{Tr}(\mathbf{A}_i \mathbf{X}))$ in problem (24) together, and discarding the constant terms, we can get the majorization problem for (24) as

$$\begin{aligned} & \underset{\mathbf{x}, \mathbf{X}}{\text{minimize}} \quad D\text{Tr}(\mathbf{X}\mathbf{X}) + 2 \sum_{i=1}^N \text{Tr}(\mathbf{X}\mathbf{A}_i) \text{Tr}(\mathbf{X}^{(k)} \mathbf{A}_i) \\ &\quad - 2D\text{Tr}(\mathbf{X}\mathbf{X}^{(k)}) - 2 \sum_{i=1}^N y_i \text{Tr}(\mathbf{A}_i \mathbf{X}) \\ & \text{subject to} \quad \mathbf{X} = \mathbf{x}\mathbf{x}^H, \end{aligned} \quad (28)$$

which is equivalent to the following leading eigenvector problem:

$$\underset{\mathbf{x}}{\text{minimize}} \quad \|\mathbf{x}\mathbf{x}^H - \mathbf{W}\|_F^2, \quad (29)$$

with the matrix

$$\mathbf{W} := \mathbf{x}^{(k)} (\mathbf{x}^{(k)})^H + \frac{1}{D} \mathbf{A} \text{diag}(\mathbf{y} - |\mathbf{A}^H \mathbf{x}^{(k)}|^2) \mathbf{A}^H. \quad (30)$$

The solution to this leading eigenvector problem is

$$\mathbf{x}^* = \sqrt{\lambda_{\max}(\mathbf{W})} \mathbf{u}_{\max}(\mathbf{W}), \quad (31)$$

where $\lambda_{\max}(\mathbf{W})$ and $\mathbf{u}_{\max}(\mathbf{W})$ are the largest eigenvalue and corresponding eigenvector of matrix \mathbf{W} . The procedures are summarized in Algorithm 3. A general choice is to conduct eigen-decomposition to find this largest eigenvalue and corresponding eigenvector, which unfortunately is usually computationally costly and time consuming. Therefore, we propose to use the power iteration method instead without conducting the eigen-decomposition. The power iteration method is a simple iterative algorithm to calculate the eigenvalue (the one with the greatest absolute value) and corresponding eigenvector. Together with the following proposition, the power iteration method will indeed produce the largest eigenvalue and the corresponding eigenvector.

Algorithm 3: PRIME-Power.

Input: $\mathbf{A}, \mathbf{y}, t_0$

1: Initial $\mathbf{x}^{(0)} \leftarrow$ leading eigenvector of $\mathbf{A} \text{diag}(\mathbf{y}) \mathbf{A}^H$

2: **for** $k = 0, \dots, t_0 - 1$ **do**

3: $\mathbf{W} = \mathbf{x}^{(k)} (\mathbf{x}^{(k)})^H + \frac{1}{D} \mathbf{A} \text{diag}(\mathbf{y} - |\mathbf{A}^H \mathbf{x}^{(k)}|^2) \mathbf{A}^H$

4: $\mathbf{x}^{(k+1)} = \sqrt{\lambda_{\max}(\mathbf{W})} \mathbf{u}_{\max}(\mathbf{W})$

5: **end for**

Output: $\mathbf{x}^{(t_0)}$.

Proposition 1: For the matrix \mathbf{W} defined in (30), its largest eigenvalue and smallest eigenvalue satisfy the inequality

$$\lambda_{\max}(\mathbf{W}) > |\lambda_{\min}(\mathbf{W})|, \quad (32)$$

provided that

$$\begin{aligned} D &> \sum_{i \in \mathcal{I}} \left(\left| \mathbf{a}_i^H \mathbf{x}^{(k)} \right|^2 - y_i \right) \frac{\|\mathbf{a}_i\|^2}{\|\mathbf{x}^{(k)}\|^2} \\ &\quad + \sum_{i=1}^N \left(\left| \mathbf{a}_i^H \mathbf{x}^{(k)} \right|^2 - y_i \right) \frac{|\mathbf{a}_i^H \mathbf{x}^{(k)}|^2}{\|\mathbf{x}^{(k)}\|^4}, \end{aligned} \quad (33)$$

where the set $\mathcal{I} := \{i : y_i < |\mathbf{a}_i^H \mathbf{x}^{(k)}|^2\}$.

Proof: Appendix B. ■

E. Convergence Analysis

Inherited from the general majorization-minimization framework, the non-increasing property (10) holds for any majorization problem. And the objective value is bounded below by 0 either for problem (2) or for problem (5). Therefore the sequence $\{f(\mathbf{x}^{(k)})\}$ generated by our algorithms is guaranteed to converge to a finite point at least. Moreover, the authors of

[20] proved that any sequence $\{\mathbf{x}^{(k)}\}$ generated by the MM algorithm converges to a stationary point when the constraint set is closed and convex. Fortunately, our three majorization problems, problem (18), problem (21), and problem (29), are unconstrained optimization problems. Therefore, according to Theorem 1 in [20]¹, the sequence $\{\mathbf{x}^{(k)}\}$ generated by our algorithms is guaranteed to converge to a stationary point of their corresponding original phase retrieval problem.

F. Computational Complexity

We now offer a short discussion on the computational complexity of all the algorithms we have proposed so far. For a general measurement matrix \mathbf{A} , the two algorithms for problem (5), namely, PRIME-Modulus-Single-Term and PRIME-Modulus-Both-Terms, both yield a simple closed-form solution and only require basic matrix multiplication at every iteration. The time complexities for these two algorithms are $O(NK)$. For problem (2), PRIME-Power needs the leading eigenvalue and corresponding eigenvector of an intermediate matrix at every iteration. The time complexity is also $O(NK)$. When the measurement matrix \mathbf{A} is from the DFT matrix, the time complexities can be reduced to $O(N \log N)$ by exploiting fast Fourier transform and inverse fast Fourier transform.

IV. ACCELERATION SCHEME

The popularity of the MM framework is attributed to its simplicity and monotonic decreasing property, at the expense of a usually low convergence rate. This slow convergence may jeopardize the performance of the MM algorithm for computing intensive tasks. In [21], the authors proposed a simple and globally convergent method to accelerate any EM algorithms. This accelerating algorithm, called the squared iterative methods (SQUAREM), generally achieves superlinear convergence, and is especially attractive in high-dimensional problems as it only requires parameter updating. Besides this, since the MM algorithm is a generalization of the EM algorithm, SQUAREM can be adopted to update the parameters in our MM-based algorithms. At every iteration, instead of updating $\mathbf{x}^{(k+1)}$ directly from $\mathbf{x}^{(k)}$, SQUAREM seeks an intermediate point \mathbf{x}' from $\mathbf{x}^{(k)}$, after which, it updates the next point $\mathbf{x}^{(k+1)}$ based on this intermediate point.

V. NUMERICAL SIMULATIONS

In this section, we present the experimental results for both problem (2) and problem (5) under various settings. Specifically, we consider that the measurement matrix is either standard complex Gaussian distributed or from the DFT matrix, and the measurements are clean or corrupted with noise. All experiments are conducted on a personal computer with a 3.20 GHz Intel Core i5-4570 CPU and 8.00 GB RAM.

¹The constraint set is limited to be in real space in [20] to guarantee differentiability. In this paper, the functions are not complex-differentiable. By introducing new variable $\tilde{\mathbf{x}} = [\text{Re}(\mathbf{x})^T, \text{Im}(\mathbf{x})^T]^T \in \mathbb{R}^{2K}$, it is easy to transform the problems into equivalent problems with real variable.

For both problems, our MM-based algorithms outperform the benchmark methods, the Wirtinger Flow algorithm and the Gerchberg-Saxton algorithm, respectively, in terms of successful recovery probability and convergence speed. Details of the experiments and comparisons can be found in later subsections under different settings.

A. Random Gaussian Matrix Setting

First we consider the case in which all the elements in the measurement matrix \mathbf{A} are independent random variables following a standard complex Gaussian distribution. Thus every element is regarded as a random variable in which the real part and the imaginary part are drawn from the standard Gaussian distribution $\mathcal{N}(0, 1)$ independently.

We choose a random signal $\mathbf{x}_o \in \mathbb{C}^{100}$ (normalized as $\mathbf{x}_o/\|\mathbf{x}_o\|$ without loss of generality) as the original signal, and generate the measurements $\mathbf{y} = |\mathbf{A}^H \mathbf{x}_o|^2 \in \mathbb{R}^N$. Since the measurement matrix \mathbf{A} is a random matrix here, we repeat the experiments 1000 times using different and independent measurement matrices, with everything else fixed as the same. In the PRIME-Power algorithm we use the power iteration method to calculate the leading eigenvalue and corresponding eigenvector instead of eigen-decomposition. Experimental results indicate that one step of the power iteration is sufficient enough to considerably reduce the computations without degrading the performance. As for the Wirtinger Flow algorithm, different from the heuristic step size used in the original paper [15], here we adopt a backtracking method to find a suitable step size. We also use SQUAREM to accelerate our algorithms, which leads to the names PRIME-Power-Acce, PRIME-Modulus-Single-Term-Acce, and PRIME-Modulus-Both-Terms-Acce accordingly.

As for the phase retrieval problem, more importance should be placed on the successful recovery probability. Therefore, as an example, we only show in Fig. 1 that our MM-based algorithms converge faster than the corresponding benchmark methods for both problems under clean measurements and the random Gaussian measurement matrix setting.

As mentioned above, all the algorithms can only recover the original signal \mathbf{x}_o up to a constant phase shift due to the loss of phase information. Fortunately, we can easily find this constant phase by the following procedure. For any solution \mathbf{x}^* returned from the algorithms, we define a function

$$h(\phi) = \|\mathbf{x}^* - \mathbf{x}_o \cdot e^{j\phi}\|_2^2. \quad (34)$$

The derivative of this function $h(\phi)$ with respect to ϕ is

$$\nabla h(\phi) = j [\mathbf{x}_o^H \mathbf{x}^* e^{-j\phi} - (\mathbf{x}^*)^H \mathbf{x}_o e^{j\phi}]. \quad (35)$$

Setting this derivative to zero, we get

$$e^{j\phi} = \frac{\mathbf{x}_o^H \mathbf{x}^*}{|\mathbf{x}_o^H \mathbf{x}^*|}. \quad (36)$$

Therefore, we can compute the squared error between the solution \mathbf{x}^* returned from our algorithms and the original signal \mathbf{x}_o , taking into consideration this global phase shift as $\|\mathbf{x}^* - \mathbf{x}_o e^{j\phi}\|_2^2$. And we plot the mean squared error (MSE) between \mathbf{x}^* and \mathbf{x}_o in Fig. 2. Besides this, for every single experiment

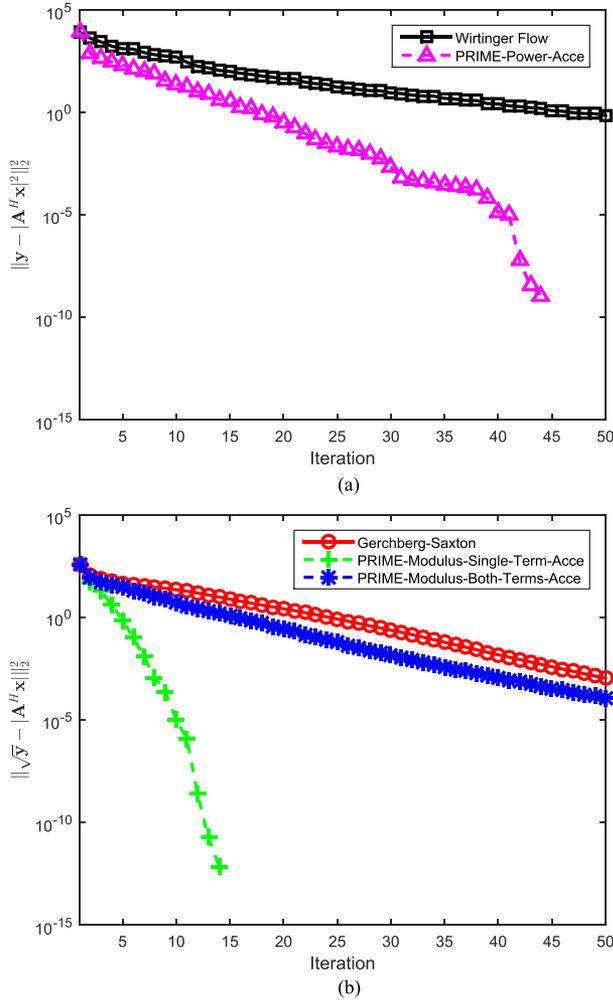


Fig. 1. Objective function versus iteration under clean measurements and random Gaussian matrix setting. $\mathbf{x} \in \mathbb{C}^{100}$, $\mathbf{y} \in \mathbb{R}^{500}$. (a) For problem (2). (b) For problem (5).

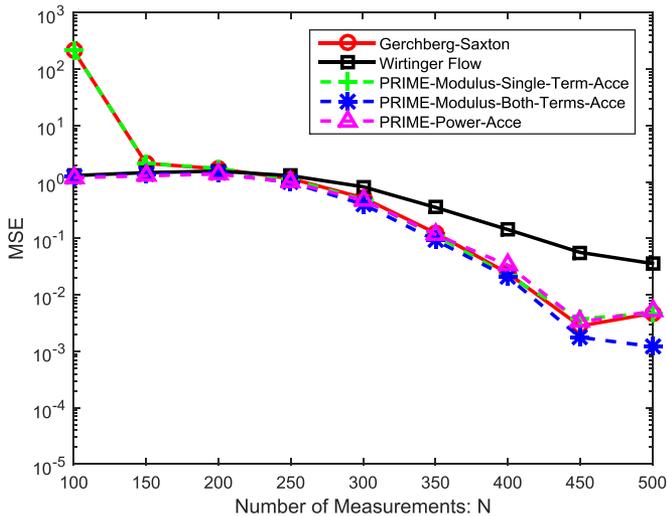


Fig. 2. Mean squared error (MSE) versus number of clean measurements under random Gaussian matrix setting. $\mathbf{x} \in \mathbb{C}^{100}$, $\mathbf{y} \in \mathbb{R}^N$.

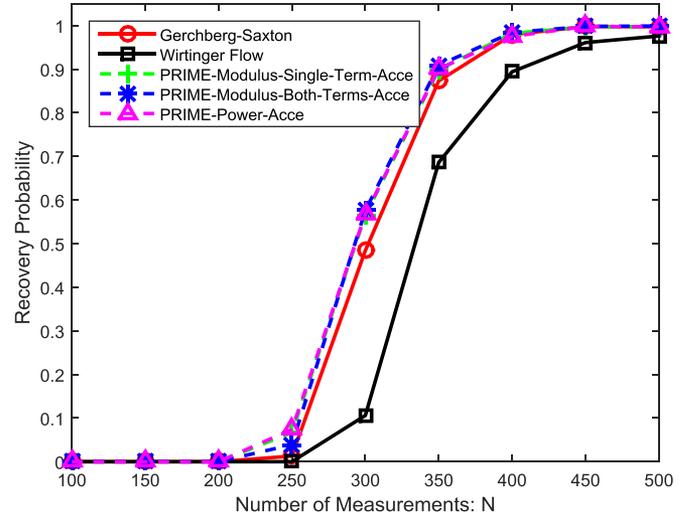


Fig. 3. Successful recovery probability versus number of clean measurements under random Gaussian matrix setting. $\mathbf{x} \in \mathbb{C}^{100}$, $\mathbf{y} \in \mathbb{R}^N$.

among these 1000 independent trials, we consider that an algorithm successfully recovers the original signal if the squared error is less than 10^{-4} . And in Fig. 3, we plot the probability of successful recovery based on these 1000 independent trails for all the algorithms.

From Figs. 2 and 3, we can see that all of our MM-based algorithms have a higher successful recovery possibility and less mean squared error than the two benchmark algorithms (PRIME-Power-Acce vs. Wirtinger Flow, PRIME-Modulus-Single-Term-Acce and PRIME-Modulus-Both-Terms-Acce vs. Gerchberg-Saxton) except PRIME-Modulus-Single-Term-Acce, which can be formulated exactly the same as the Gerchberg-Saxton algorithm when not accelerated. And it agrees with the conjecture in [8] that about $4K$ measurements are needed for a successful recovery with high probability.

B. Discrete Fourier Transform Matrix Setting

In traditional phase retrieval problems, the measurements are the magnitude of the Fourier transform of the signal. Hence, in this subsection, we consider that the measurement matrix \mathbf{A} is from the DFT matrix; i.e., \mathbf{A}^H is equal to the matrix that consists of the first K columns of the $N \times N$ DFT matrix. There are certain advantages to using the DFT properties in our majorization problems. First of all, the leading eigenvalue of the matrix Φ is as easy as $\lambda_{\max}(\Phi) = NK$ (proof in Appendix C). And the leading eigenvalue needed in problem (21) also has a simple form now $\lambda_{\max}(\mathbf{A}\mathbf{A}^H) = N$ (proof in Appendix A).

Note that there are some differences between the DFT matrix setting and the former random Gaussian matrix setting. The measurement matrix \mathbf{A} is now from the DFT matrix and is not random anymore. The only randomness comes from the original signal \mathbf{x}_o . Therefore we need to use different original signals in the 1000 trials. Another difference is that there are more ambiguities under the DFT matrix setting, unlike under the random Gaussian matrix setting where the global constant phase shift is the only ambiguity. The authors of [14] pointed out that there are

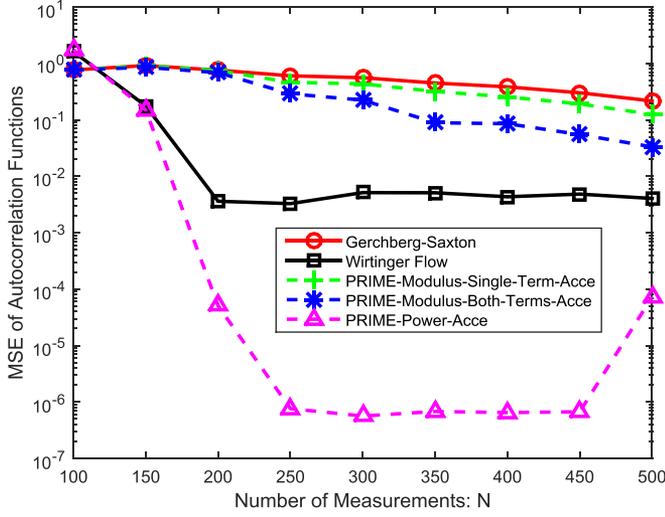


Fig. 4. Mean squared error (MSE) of autocorrelation functions versus number of clean measurements under DFT matrix setting. $\mathbf{x} \in \mathbb{C}^{100}$, $\mathbf{y} \in \mathbb{R}^N$.

always trivial ambiguities and non-trivial ambiguities under the DFT matrix setting for a one dimensional signal. For the trivial ambiguities, any individual or combination of the following three transformations conserve the Fourier magnitude:

1. Global constant phase shift: $\mathbf{x} \rightarrow \mathbf{x} \cdot e^{j\phi}$,
2. Circular shift: $[\mathbf{x}]_i \rightarrow [\mathbf{x}]_{(i+i_0) \bmod K}$,
3. Conjugate inversion: $[\mathbf{x}]_i \rightarrow [\mathbf{x}]_{K-i}$.

As for the non-trivial ambiguities, any two signals which have the same autocorrelation function share the same Fourier magnitude. Actually, any two signals within the trivial ambiguities also yield the same autocorrelation function. Therefore under the DFT matrix setting, we can only recover the signal up to the same autocorrelation function without additional information. We use the following autocorrelation function:

$$[\mathbf{r}]_m = \sum_{i=\max\{1, m+1\}}^K [\mathbf{x}]_i \overline{[\mathbf{x}]_{i-m}}, \quad m = -(K-1), \dots, K-1. \quad (37)$$

First, we calculate the autocorrelation function of the original signal \mathbf{r}_o and the autocorrelation function of the solution returned from our algorithms \mathbf{r}^* . Later on, we compute the squared error between these two autocorrelation functions $\|\mathbf{r}_o - \mathbf{r}^*\|_2^2$. We also repeat the experiment 1000 times with different and independent original signals \mathbf{x}_o .

In Fig. 4, we plot the mean squared error of the autocorrelation functions over these 1000 independent trials, and in Fig. 5, we plot the probability for successful recovery based on these 1000 experiments. In every experiment, an algorithm is considered to successfully recover the signal if the squared error $\|\mathbf{r}_o - \mathbf{r}^*\|_2^2$ is less than 10^{-4} . As shown in Figs. 4 and 5, all of our MM-based algorithms have less mean squared error of the autocorrelation function and successfully recover the signal with a higher probability than the corresponding benchmark algorithms. Furthermore, the differences between our algorithms and the corresponding benchmark algorithms are significantly larger than those in the random Gaussian case. And it agrees with our

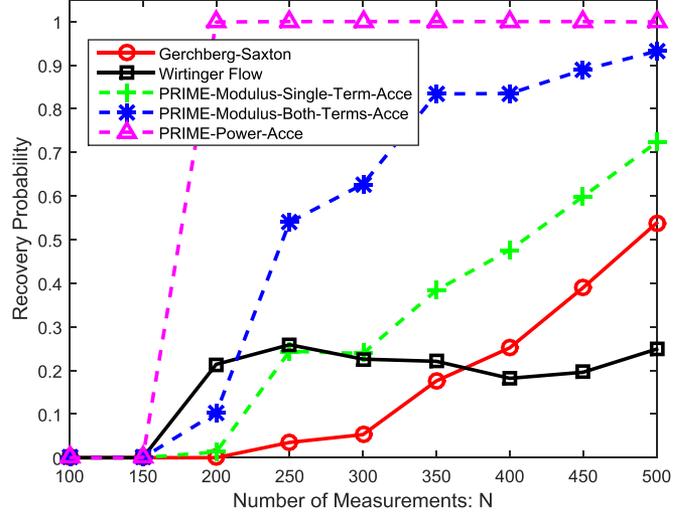


Fig. 5. Successful recovery probability of autocorrelation functions versus number of clean measurements under DFT matrix setting. $\mathbf{x} \in \mathbb{C}^{100}$, $\mathbf{y} \in \mathbb{R}^N$.

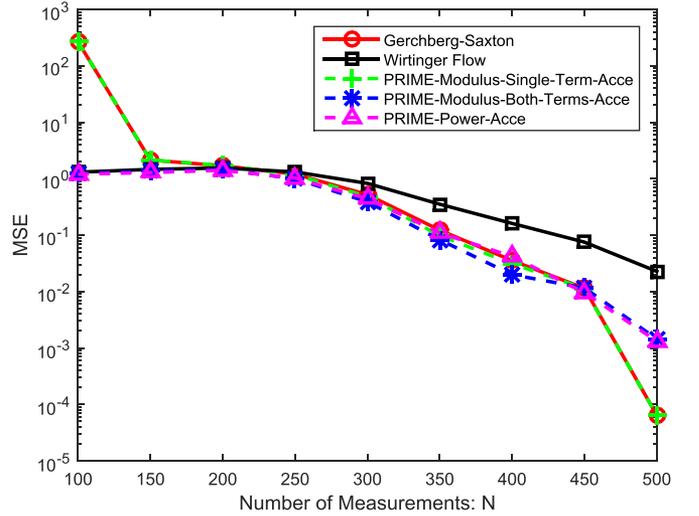


Fig. 6. Mean squared error (MSE) versus number of noisy measurements under random Gaussian matrix setting. Add noise to \mathbf{y} . $\mathbf{x} \in \mathbb{C}^{100}$, $\mathbf{y} \in \mathbb{R}^N$.

former analysis that it is beneficial to majorize both terms for problem (5) under the DFT matrix setting. PRIME-Modulus-Both-Terms-AcCe outperforms PRIME-Modulus-Single-Term-AcCe in terms of successful recovery probability and mean squared error.

C. Robustness to Noise

Up to now the experimental results agree with our theoretic analysis that our algorithms outperform the benchmark algorithms under the clean measurements setting. However, in real life the measurements are always corrupted with noise, and usually noise will degrade the performance of an algorithm. Therefore it is necessary to take the noise into consideration. In this subsection, we present the results when the measurements are corrupted with noise.

First, we consider adding random Gaussian noise to the measurements $\mathbf{y} = |\mathbf{A}^H \mathbf{x}_o|^2 + \mathbf{n}$. The same noisy measurements

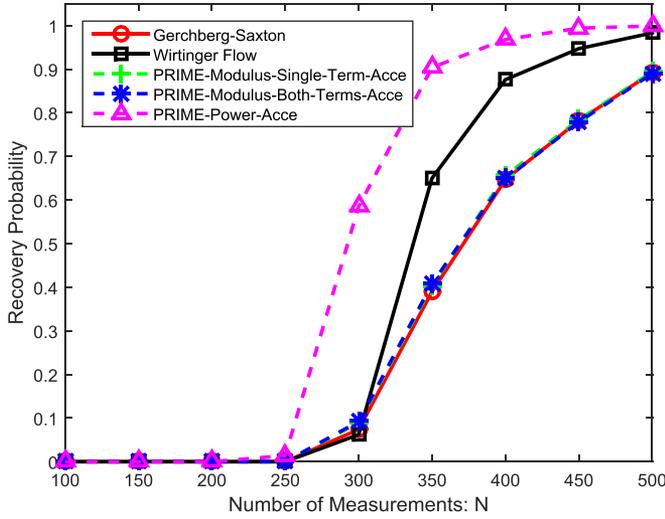


Fig. 7. Successful recovery probability versus number of noisy measurements under random Gaussian matrix setting. Add noise to \mathbf{y} . $\mathbf{x} \in \mathbb{C}^{100}$, $\mathbf{y} \in \mathbb{R}^N$.

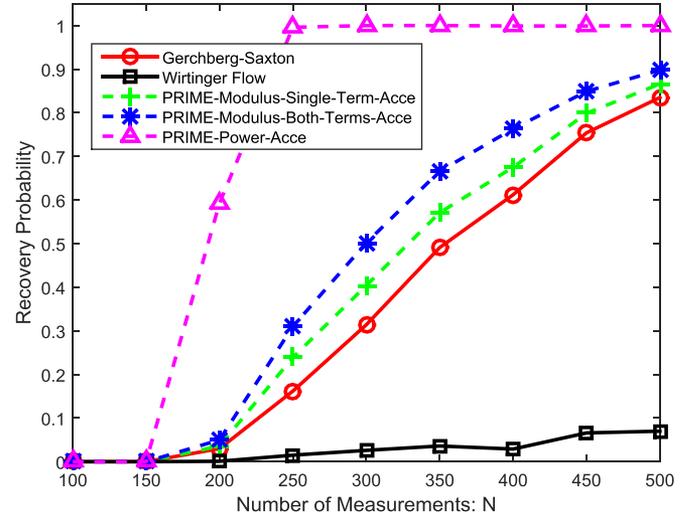


Fig. 9. Successful recovery probability of autocorrelation functions versus number of noisy measurements under DFT matrix setting. Add noise to \mathbf{y} . $\mathbf{x} \in \mathbb{C}^{100}$, $\mathbf{y} \in \mathbb{R}^N$.

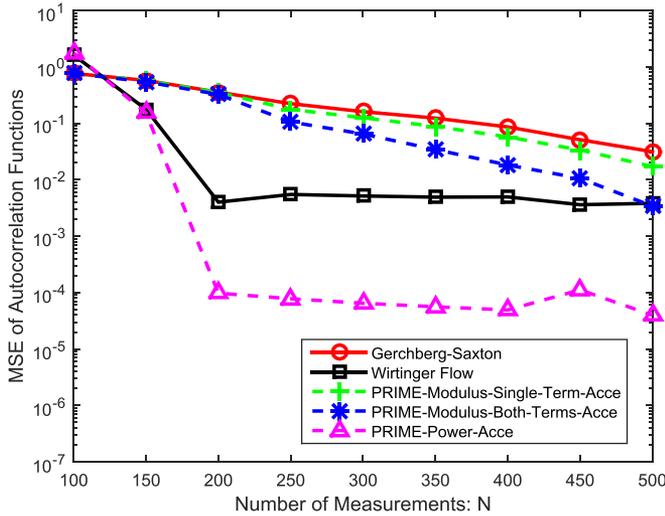


Fig. 8. Mean squared error (MSE) of autocorrelation functions versus number of noisy measurements under DFT matrix setting. Add noise to \mathbf{y} . $\mathbf{x} \in \mathbb{C}^{100}$, $\mathbf{y} \in \mathbb{R}^N$.

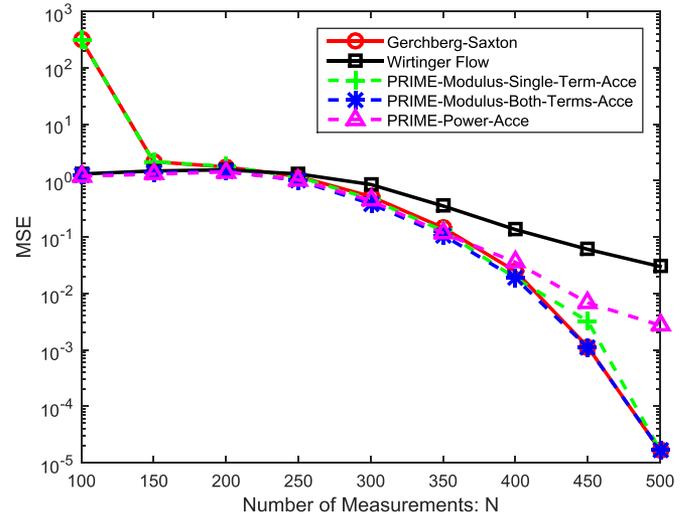


Fig. 10. Mean squared error (MSE) versus number of noisy measurements under random Gaussian matrix setting. Add noise to $\sqrt{\mathbf{y}}$. $\mathbf{x} \in \mathbb{C}^{100}$, $\mathbf{y} \in \mathbb{R}^N$.

\mathbf{y} are provided for both problems (2) and (5). We repeat the experiments 1000 times under the same settings as in the clean measurements case. The signal-to-noise ratio (SNR) is about 23 dB for the random Gaussian matrix setting and 20 dB for the DFT matrix setting. Results of the mean squared error and successful recovery probability are presented in Figs. 6 and 7 for the random Gaussian matrix setting and Figs. 8 and 9 for the DFT matrix setting.

Different from problem (2) where the modulus squared measurements \mathbf{y} are exploited, problem (5) uses the modulus information $\sqrt{\mathbf{y}}$ instead. As a fair comparison, we also need to consider adding noise to the modulus information $\sqrt{\mathbf{y}} = |\mathbf{A}^H \mathbf{x}_o| + \mathbf{n}$. The same noisy modulus information $\sqrt{\mathbf{y}}$ is provided for problems (2) and (5). Again, we repeat the experiments 1000 times under the same rules as in the clean measurements case. The SNR is also about 23 dB for the random

Gaussian matrix setting and 20 dB for the DFT matrix setting. Results are shown in Figs. 10, 11, 12 and 13, respectively.

Comparing the successful recovery probabilities (Figs. 3, 7, and 11 for the random Gaussian matrix setting, and Figs. 5, 9, and 13 for the DFT matrix setting), we find out that under the Gaussian matrix setting, noisy measurements on \mathbf{y} dramatically decrease the performance of algorithms for problem (5) (Figs. 3 and 7). On the other hand, the performance of all algorithms is almost the same as in the clean measurements case when noise pollutes the modulus information $\sqrt{\mathbf{y}}$ (Figs. 3 and 11). Under the DFT matrix setting, the performance is almost the same in both noisy circumstances (Figs. 9 and 13). Surprisingly, while the noise slightly decreases the performance of PRIME-Power-Acce, Wirtinger Flow, and PRIME-Modulus-Both-Terms-Acce, it significantly improves the recovery probability of the other two algorithms for problem (5) (Figs. 5, 9, and 13). The reason is

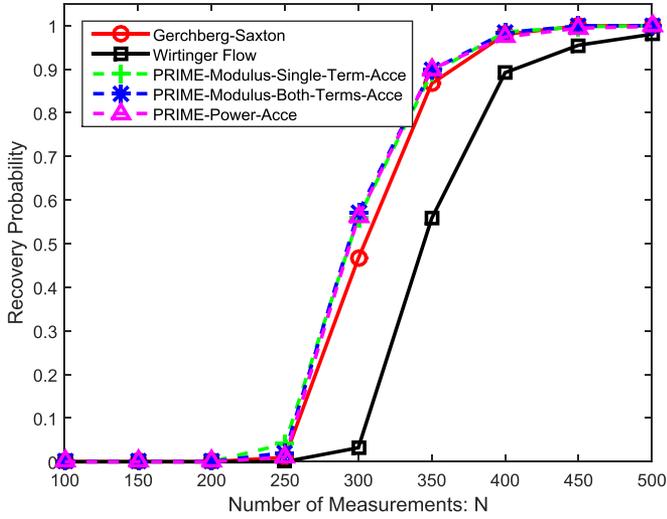


Fig. 11. Successful recovery probability versus number of noisy measurements under random Gaussian matrix setting. Add noise to \sqrt{y} . $x \in \mathbb{C}^{100}$, $y \in \mathbb{R}^N$.

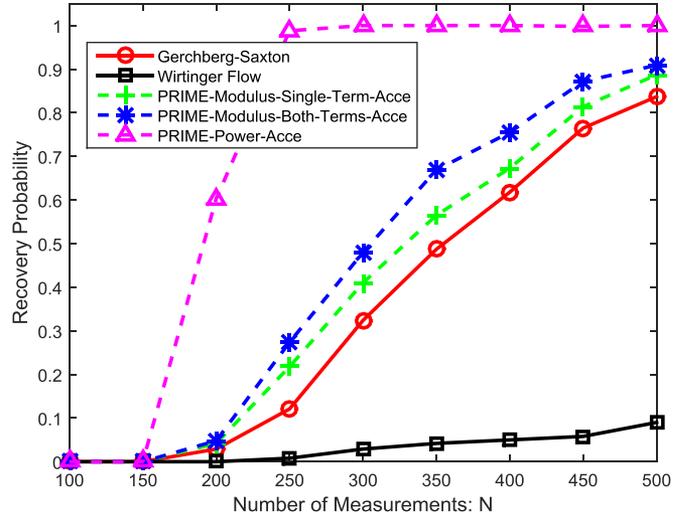


Fig. 13. Successful recovery probability of autocorrelation functions versus number of noisy measurements under DFT matrix setting. Add noise to \sqrt{y} . $x \in \mathbb{C}^{100}$, $y \in \mathbb{R}^N$.

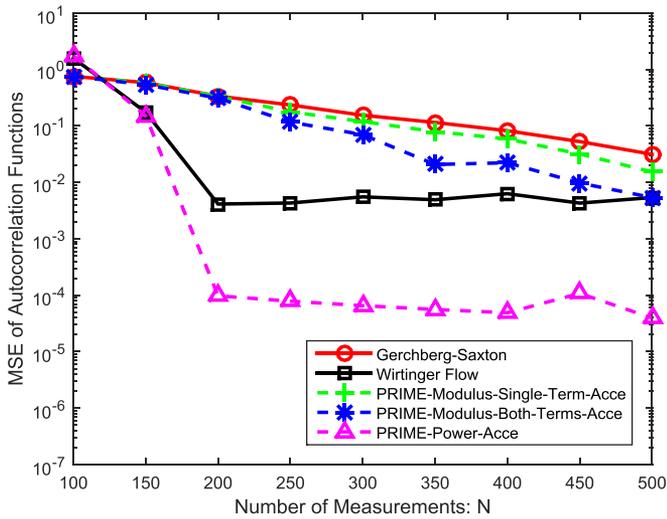


Fig. 12. Mean squared error (MSE) of autocorrelation functions versus number of noisy measurements under DFT matrix setting. Add noise to \sqrt{y} . $x \in \mathbb{C}^{100}$, $y \in \mathbb{R}^N$.

that the “dithering” effect of noise may help these two algorithms get rid of some bad stationary points.

As for the mean squared error (Figs. 2, 6, and 10 for the random Gaussian matrix setting, and Figs. 4, 8, and 12 for the DFT matrix setting), the plots are almost the same. This is because the values of the mean squared error are dominated by those experiments with unsuccessful recoveries, which usually have a significantly large squared error. And adding small noise cannot change an unsuccessful recovery to a successful one in most cases.

Among all these figures, our MM-based algorithms outperform their corresponding benchmark methods in terms of both successful recovery probability and mean squared error under various settings. And PRIME-Power-AcCe turns out to be the best algorithm since it has the highest successful recovery

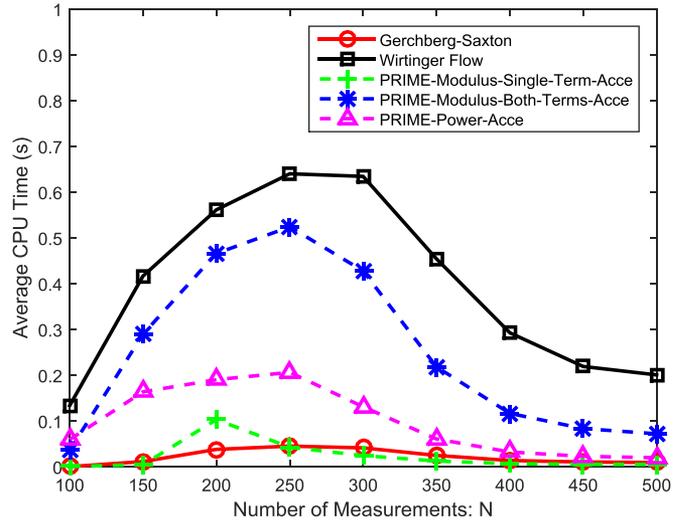


Fig. 14. Average CPU time versus number of clean measurements under random Gaussian matrix setting. $x \in \mathbb{C}^{100}$, $y \in \mathbb{R}^N$.

probability under all different settings, and it is also more robust to noise. Its advantages over other algorithms are more significant under DFT matrix setting.

In Fig. 14, we plot the average CPU time for all the algorithms over 1000 Monte Carlo experiments under clean measurements and the random Gaussian measurement matrix setting. For problem (2), PRIME-Power-AcCe takes much less time than the Wirtinger Flow algorithm. For problem (5), PRIME-Modulus-Single-Term-AcCe takes about same time as the Gerchberg-Saxton algorithm. PRIME-Modulus-Both-Terms-AcCe needs more time because of its slow convergence, but it outperforms the other two algorithms significantly under the DFT matrix setting.

Finally in Fig. 15, we demonstrate practical image recovery under noisy measurements and the random Gaussian matrix setting. The modulus information \sqrt{y} are contaminated

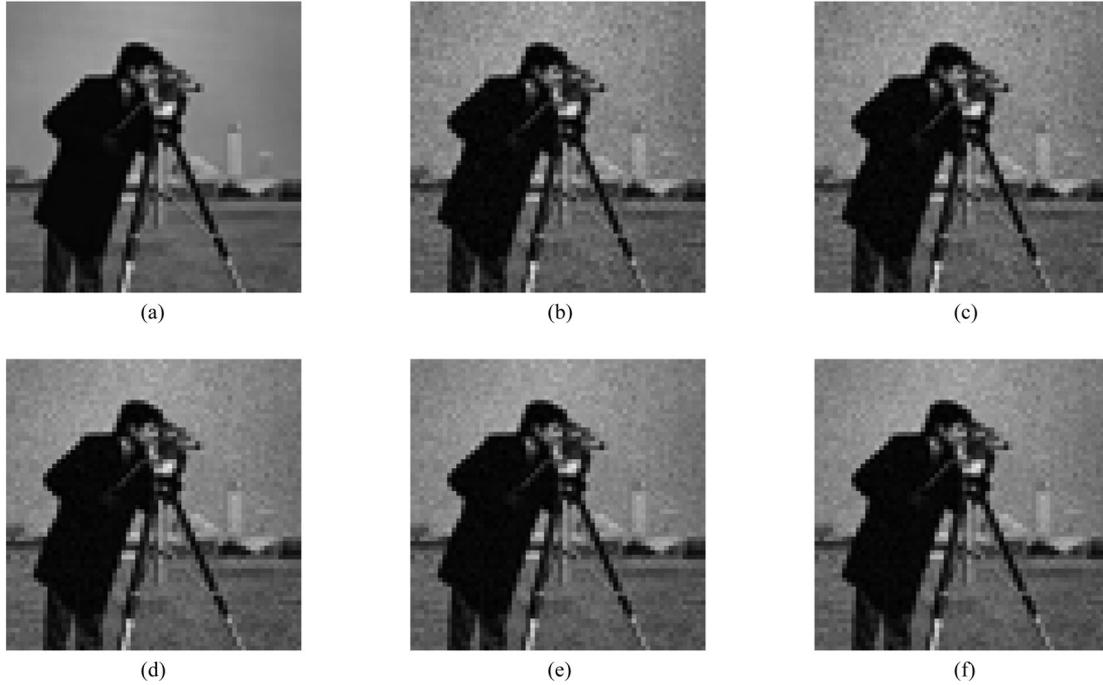


Fig. 15. Practical image recovery under noisy measurements and random Gaussian matrix setting. Add noise to \sqrt{y} , SNR = 12 dB. The original cameraman image is of size 64×64 , i.e., $K = 64^2 = 4096$ and $N = 4K$. (a) Original, (b) Wirtinger Flow NMSE = -25.8 dB, $t = 9557.1$ s, (c) PRIME-Power-Acce NMSE = -25.8 dB, $t = 376.6$ s, (d) Gerchberg-Saxton NMSE = -27.7 dB, $t = 293.9$ s, (e) PRIME-Modulus-Single-Term-Acce NMSE = -27.7 dB, $t = 122.5$ s, (f) PRIME-Modulus-Both-Terms-Acce NMSE = -27.7 dB, $t = 2532.1$ s.

by random Gaussian noise with SNR about 12 dB. For problem (2), PRIME-Power-Acce is much faster than the Wirtinger Flow algorithm (about 25 times faster). And they return the same solution (same normalized mean squared error (NMSE)). For problem (5), the corresponding three algorithms also yield the same solution. PRIME-Modulus-Single-Term-Acce takes less than half the time of the Gerchberg-Saxton algorithm. But PRIME-Modulus-Both-Terms-Acce needs more time because of its slow convergence.

VI. CONCLUSION

Phase retrieval is of great interest in physics and engineering. It is not a linear recovery problem due to the loss of phase information. Algorithms based on semidefinite relaxation manage to recover the original signal by solving a convex semidefinite programming problem. But they are not applicable to large scale problems because of the dimension increase in the matrix-lifting procedure. The Wirtinger Flow algorithm recovers the original signal from the modulus squared of its linear measurements (problem (2)) using the gradient descent method, but the performance is relatively poor. The classical Gerchberg-Saxton algorithm recovers the original signal from the modulus of its linear measurements (problem (5)) through alternating minimizations by introducing a new variable representing phase information. In this paper we have proposed three efficient algorithms under the majorization-minimization framework that outperform the existing methods in terms of successful recovery probability and mean squared error. Instead of dealing with the cumbersome

phase retrieval problems directly, we have considered different majorization problems which yield a simple closed-form solution via different majorization-minimization techniques. Theoretic analysis as well as experimental results under various settings are also presented in the paper to further validate the efficiency of our algorithms.

APPENDIX A

PROOF OF $\lambda_{\max}(\mathbf{A}\mathbf{A}^H) = N$ FOR DFT MATRIX

The elements in the DFT measurement matrix $\mathbf{A} \in \mathbb{C}^{K \times N}$ ($K \leq N$) are

$$A_{ki} = e^{j \frac{2\pi(k-1)(i-1)}{N}}, \quad k = 1, \dots, K, \text{ and } i = 1, \dots, N. \quad (38)$$

Hence the element at the m -th row and n -th column of the square matrix $\mathbf{A}\mathbf{A}^H \in \mathbb{C}^{K \times K}$ is

$$\begin{aligned} [\mathbf{A}\mathbf{A}^H]_{mn} &= \sum_{k=1}^N A_{mk} \overline{A_{nk}} \\ &= \sum_{k=1}^N e^{j \frac{2\pi(m-1)(k-1)}{N}} e^{-j \frac{2\pi(n-1)(k-1)}{N}} \\ &= \sum_{k=1}^N e^{j \frac{2\pi(m-n)(k-1)}{N}} = \begin{cases} N, & m = n, \\ 0, & \text{otherwise.} \end{cases} \quad (39) \end{aligned}$$

Thus $\mathbf{A}\mathbf{A}^H = N\mathbf{I}_K$. Therefore $\lambda_{\max}(\mathbf{A}\mathbf{A}^H) = N$.

APPENDIX B
 PROOF OF PROPOSITION 1

First,

$$\begin{aligned} \lambda_{\max}(\mathbf{W}) &\geq \frac{(\mathbf{x}^{(k)})^H \mathbf{W} \mathbf{x}^{(k)}}{\|\mathbf{x}^{(k)}\|} = \|\mathbf{x}^{(k)}\|^2 \\ &+ \frac{1}{D} \sum_{i=1}^N \left(y_i - |\mathbf{a}_i^H \mathbf{x}^{(k)}|^2 \right) \frac{|\mathbf{a}_i^H \mathbf{x}^{(k)}|^2}{\|\mathbf{x}^{(k)}\|^2}. \end{aligned} \quad (40)$$

If $\mathcal{I} = \emptyset$, \mathbf{W} is positive semidefinite, and it is trivial that $\lambda_{\max}(\mathbf{W}) > \lambda_{\min}(\mathbf{W}) \geq 0$. When $\mathcal{I} \neq \emptyset$ and \mathbf{W} is not positive semidefinite, defining matrix

$$\mathbf{Z} := \mathbf{A} \text{diag} \left(\mathbf{y} - \left| \mathbf{A}^H \mathbf{x}^{(k)} \right|^2 \right) \mathbf{A}, \quad (41)$$

then

$$\frac{1}{D} \lambda_{\min}(\mathbf{Z}) \leq \lambda_{\min}(\mathbf{W}) < 0. \quad (42)$$

Therefore, $\lambda_{\max}(\mathbf{W}) > |\lambda_{\min}(\mathbf{W})|$ will hold if

$$\begin{aligned} \|\mathbf{x}^{(k)}\|^2 + \frac{1}{D} \sum_{i=1}^N \left(y_i - |\mathbf{a}_i^H \mathbf{x}^{(k)}|^2 \right) \frac{|\mathbf{a}_i^H \mathbf{x}^{(k)}|^2}{\|\mathbf{x}^{(k)}\|^2} \\ > -\frac{\lambda_{\min}(\mathbf{Z})}{D}, \end{aligned} \quad (43)$$

which is equivalent to

$$D > -\frac{\lambda_{\min}(\mathbf{Z})}{\|\mathbf{x}^{(k)}\|^2} + \sum_{i=1}^N \left(\left| \mathbf{a}_i^H \mathbf{x}^{(k)} \right|^2 - y_i \right) \frac{|\mathbf{a}_i^H \mathbf{x}^{(k)}|^2}{\|\mathbf{x}^{(k)}\|^4}. \quad (44)$$

Note that

$$-\lambda_{\min}(\mathbf{Z}) = \lambda_{\max}(-\mathbf{Z}) \leq \sum_{i \in \mathcal{I}} \left(\left| \mathbf{a}_i^H \mathbf{x}^{(k)} \right|^2 - y_i \right) \|\mathbf{a}_i\|^2. \quad (45)$$

Therefore, $\lambda_{\max}(\mathbf{W}) > |\lambda_{\min}(\mathbf{W})|$ will hold if

$$\begin{aligned} D > \sum_{i \in \mathcal{I}} \left(\left| \mathbf{a}_i^H \mathbf{x}^{(k)} \right|^2 - y_i \right) \frac{\|\mathbf{a}_i\|^2}{\|\mathbf{x}^{(k)}\|^2} \\ + \sum_{i=1}^N \left(\left| \mathbf{a}_i^H \mathbf{x}^{(k)} \right|^2 - y_i \right) \frac{|\mathbf{a}_i^H \mathbf{x}^{(k)}|^2}{\|\mathbf{x}^{(k)}\|^4}. \end{aligned} \quad (46)$$

 APPENDIX C
 PROOF OF $\lambda_{\max}(\Phi) = NK$ FOR DFT MATRIX

Recall the definition of the Hermitian matrix

$$\mathbf{A}_i = \mathbf{a}_i \mathbf{a}_i^H \in \mathbb{C}^{K \times K}, \quad i = 1, \dots, N, \quad K \leq N. \quad (47)$$

Hence the element at the m -th row and n -th column of this square matrix is

$$\begin{aligned} [\mathbf{A}_i]_{mn} &= [\mathbf{a}_i]_m \cdot \overline{[\mathbf{a}_i]_n} = e^{j \frac{2\pi(i-1)(m-1)}{N}} e^{-j \frac{2\pi(i-1)(n-1)}{N}} \\ &= e^{j \frac{2\pi(i-1)(m-n)}{N}}, \quad i = 1, \dots, N, \quad \text{and } m, n \\ &= 1, \dots, K. \end{aligned} \quad (48)$$

So the $((s-1)K + t)$ -th element in the vector $\text{vec}(\mathbf{A}_i)$ is

$$\begin{aligned} [\text{vec}(\mathbf{A}_i)]_{(s-1)K+t} &= [\mathbf{A}_i]_{ts} \\ &= e^{j \frac{2\pi(i-1)(t-s)}{N}}, \quad t, s = 1, \dots, K. \end{aligned} \quad (49)$$

Also recall the definition of the Hermitian matrix

$$\Phi = \sum_{i=1}^N \text{vec}(\mathbf{A}_i) \text{vec}(\mathbf{A}_i)^H \in \mathbb{C}^{K^2 \times K^2}. \quad (50)$$

Thus the element at the $((s_1-1)K + t_1)$ -th row and $((s_2-1)K + t_2)$ -th column of matrix Φ is

$$\begin{aligned} [\Phi]_{(s_1-1)K+t_1, (s_2-1)K+t_2} &= \sum_{i=1}^N e^{j \frac{2\pi(i-1)(t_1-s_1)}{N}} e^{-j \frac{2\pi(i-1)(t_2-s_2)}{N}} \\ &= \sum_{i=1}^N e^{j \frac{2\pi(i-1)(t_1-s_1-t_2+s_2)}{N}} \\ &= \begin{cases} N, & t_1 - s_1 = t_2 - s_2, \\ 0, & \text{otherwise,} \end{cases} \\ & \quad t_1, t_2, s_1, s_2 = 1, \dots, K. \end{aligned} \quad (51)$$

The summation of all the elements at the $((s_1-1)K + t_1)$ -th row of the matrix Φ is

$$[\Phi \cdot \mathbf{1}]_{(s_1-1)K+t_1} = \sum_{s_2=1}^K \sum_{t_2=t_1-s_1+s_2}^K N \leq NK, \quad (52)$$

where equality is achieved when $s_1 = t_1$.

Note that the matrix Φ is a symmetric matrix with each element either N or 0 . And it is also positive semidefinite by the definition. Therefore all the eigenvalues of the matrix Φ are nonnegative real numbers. For any vector $\mathbf{x} \in \mathbb{C}^{K^2}$,

$$\begin{aligned} \mathbf{x}^H (NK\mathbf{I} - \Phi) \mathbf{x} &\geq \mathbf{x}^H (\text{diag}(\Phi \cdot \mathbf{1}) - \Phi) \mathbf{x} \\ &= \sum_{m=1}^{K^2} \overline{x_m} x_m \sum_{n=1}^{K^2} [\Phi]_{mn} - \sum_{m=1}^{K^2} \sum_{n=1}^{K^2} \overline{x_m} [\Phi]_{mn} x_n \\ &= \sum_{m=1}^{K^2} \sum_{n=1}^{K^2} [\Phi]_{mn} \overline{x_m} (x_m - x_n) \\ &= \frac{1}{2} \sum_{m=1}^{K^2} \sum_{n=1}^{K^2} [\Phi]_{mn} [\overline{x_m} (x_m - x_n) + \overline{x_n} (x_n - x_m)] \\ &= \frac{1}{2} \sum_{m=1}^{K^2} \sum_{n=1}^{K^2} [\Phi]_{mn} |x_m - x_n|^2 \geq 0, \end{aligned} \quad (53)$$

where the third equality comes from the fact that Φ is a symmetric real matrix. Therefore,

$$\lambda_{\max}(\Phi) \leq NK. \quad (54)$$

Now we choose $\mathbf{x} = \text{vec}(\mathbf{I}_K)$. Then

$$\begin{aligned} \frac{\mathbf{x}^H \Phi \mathbf{x}}{\mathbf{x}^H \mathbf{x}} &= \sum_{i=1}^N \frac{\text{vec}(\mathbf{I}_K)^H \text{vec}(\mathbf{A}_i) \text{vec}(\mathbf{A}_i)^H \text{vec}(\mathbf{I}_K)}{\text{vec}(\mathbf{I}_K)^H \text{vec}(\mathbf{I}_K)} \\ &= \sum_{i=1}^N \frac{(\text{Tr}(\mathbf{A}_i))^2}{\text{Tr}}(\mathbf{I}_K) = N \frac{K^2}{K} = NK. \end{aligned} \quad (55)$$

Therefore the leading eigenvalue $\lambda_{\max}(\Phi) = NK$.

REFERENCES

- [1] A. Walther, "The question of phase retrieval in optics," *Optica Acta: Int. J. Opt.*, vol. 10, no. 1, pp. 41–49, 1963. [Online]. Available: <http://dx.doi.org/10.1080/713817747>.
- [2] R. W. Harrison, "Phase problem in crystallography," *J. Opt. Soc. Amer. A*, vol. 10, no. 5, pp. 1046–1055, May 1993. [Online]. Available: <http://josaa.osa.org/abstract.cfm?URI=josaa-10-5-1046>
- [3] J. Miao, T. Ishikawa, B. Johnson, E. H. Anderson, B. Lai, and K. O. Hodgson, "High resolution 3d x-ray diffraction microscopy," *Phys. Rev. Lett.*, vol. 89, p. 088303, Aug. 2002. [Online]. Available: <http://link.aps.org/doi/10.1103/PhysRevLett.89.088303>
- [4] N. Sturmel and L. Daudet, "Signal reconstruction from STFT magnitude: A state of the art," in *Proc. Int. Conf. Digit. Audio Effects DAFX*, vol. 2012, 2011, pp. 375–386.
- [5] J. Le Roux and E. Vincent, "Consistent Wiener filtering for audio source separation," *IEEE Signal Process. Lett.*, vol. 20, no. 3, pp. 217–220, Mar. 2013.
- [6] T. Gerkmann, M. Krawczyk-Becker, and J. Le Roux, "Phase processing for single-channel speech enhancement: History and recent advances," *IEEE Signal Process. Mag.*, vol. 32, no. 2, pp. 55–66, Mar. 2015.
- [7] E. J. Candès, T. Strohmer, and V. Voroninski, "Phaselift: Exact and stable signal recovery from magnitude measurements via convex programming," *Commun. Pure Appl. Math.*, vol. 66, no. 8, pp. 1241–1274, 2013.
- [8] A. S. Bandeira, J. Cahill, D. G. Mixon, and A. A. Nelson, "Saving phase: Injectivity and stability for phase retrieval," *Appl. Comput. Harmon. Anal.*, vol. 37, no. 1, pp. 106–125, 2014. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S1063520313000936>.
- [9] R. W. Gerchberg and W. O. Saxton, "A practical algorithm for the determination of phase from image and diffraction plane pictures," *Optik*, vol. 35, p. 237, 1972.
- [10] J. R. Fienup, "Reconstruction of an object from the modulus of its Fourier transform," *Opt. Lett.*, vol. 3, no. 1, pp. 27–29, Jul. 1978. [Online]. Available: <http://ol.osa.org/abstract.cfm?URI=ol-3-1-27>
- [11] P. Netrapalli, P. Jain, and S. Sanghavi, "Phase retrieval using alternating minimization," *IEEE Trans. Signal Process.*, vol. 63, no. 18, pp. 4814–4826, Sep. 2015.
- [12] I. Waldspurger, A. d'Asspremont, and S. Mallat, "Phase recovery, Maxcut and complex semidefinite programming," *Math. Program.*, vol. 149, no. 1–2, pp. 47–81, 2015. [Online]. Available: <http://dx.doi.org/10.1007/s10107-013-0738-9>.
- [13] E. Candès, Y. Eldar, T. Strohmer, and V. Voroninski, "Phase retrieval via matrix completion," *SIAM J. Imag. Sci.*, vol. 6, no. 1, pp. 199–225, 2013. [Online]. Available: <http://dx.doi.org/10.1137/110848074>.
- [14] Y. Shechtman, Y. Eldar, O. Cohen, H. Chapman, J. Miao, and M. Segev, "Phase retrieval with application to optical imaging: A contemporary overview," *IEEE Signal Process. Mag.*, vol. 32, no. 3, pp. 87–109, May 2015.
- [15] E. Candès, X. Li, and M. Soltanolkotabi, "Phase retrieval via Wirtinger Flow: Theory and algorithms," *IEEE Trans. Inf. Theory*, vol. 61, no. 4, pp. 1985–2007, Apr. 2015.
- [16] Y. Shechtman, A. Beck, and Y. Eldar, "GESPAR: Efficient phase retrieval of sparse signals," *IEEE Trans. Signal Process.*, vol. 62, no. 4, pp. 928–938, Feb. 2014.
- [17] P. Schniter and S. Rangan, "Compressive phase retrieval via generalized approximate message passing," *IEEE Trans. Signal Process.*, vol. 63, no. 4, pp. 1043–1055, Feb. 2015.
- [18] D. R. Hunter and K. Lange, "A tutorial on MM algorithms," *Amer. Statist.*, vol. 58, no. 1, pp. 30–37, 2004. [Online]. Available: <http://dx.doi.org/10.1198/0003130042836>.
- [19] J. Song, P. Babu, and D. Palomar, "Optimization methods for designing sequences with low autocorrelation sidelobes," *IEEE Trans. Signal Process.*, vol. 63, no. 15, pp. 3998–4009, Aug. 2015.
- [20] M. Razaviyayn, M. Hong, and Z.-Q. Luo, "A unified convergence analysis of block successive minimization methods for nonsmooth optimization," *SIAM J. Optim.*, vol. 23, no. 2, pp. 1126–1153, 2013. [Online]. Available: <http://dx.doi.org/10.1137/120891009>.
- [21] R. Varadhan and C. Roland, "Simple and globally convergent methods for accelerating the convergence of any EM algorithm," *Scand. J. Statist.*, vol. 35, no. 2, pp. 335–353, 2008. [Online]. Available: <http://dx.doi.org/10.1111/j.1467-9469.2007.00585.x>.

Tianyu Qiu received the B.Eng. degree in electronic and information from the Huazhong University of Science and Technology, Wuhan, China, in 2013. He is currently pursuing the Ph.D. degree in electronic and computer engineering at The Hong Kong University of Science and Technology. His research interests include statistical signal processing, mathematical optimization, and machine learning.

Prabhu Babu received the Ph.D. degree in electrical engineering from the Uppsala University, Sweden, in 2012.



Daniel P. Palomar (S'99–M'03–SM'08–F'12) received the electrical engineering and Ph.D. degrees from the Technical University of Catalonia (UPC), Barcelona, Spain, in 1998 and 2003, respectively.

He is a Professor in the Department of Electronic and Computer Engineering at the Hong Kong University of Science and Technology (HKUST), Hong Kong, which he joined in 2006. Since 2013 he is a Fellow of the Institute for Advance Study (IAS) at HKUST. He had previously held several research appointments, namely, at King's College London (KCL), London, U.K.; Stanford University, Stanford, CA; Telecommunications Technological Center of Catalonia (CTTC), Barcelona, Spain; Royal Institute of Technology (KTH), Stockholm, Sweden; University of Rome "La Sapienza," Rome, Italy; and Princeton University, Princeton, NJ. His current research interests include applications of convex optimization theory, game theory, and variational inequality theory to financial systems, big data systems, and communication systems.

Dr. Palomar is a recipient of a 2004–2006 Fulbright Research Fellowship, the 2004 and 2015 (co-author) Young Author Best Paper Awards by the IEEE Signal Processing Society, the 2015–2016 HKUST Excellence Research Award, the 2002–2003 best Ph.D. prize in Information Technologies and Communications by the Technical University of Catalonia (UPC), the 2002–2003 Rosina Ribalta first prize for the Best Doctoral Thesis in Information Technologies and Communications by the Epsom Foundation, and the 2004 prize for the best Doctoral Thesis in Advanced Mobile Communications by the Vodafone Foundation and COIT.

He is a Guest Editor of the IEEE JOURNAL OF SELECTED TOPICS IN SIGNAL PROCESSING 2016 Special Issue on Financial Signal Processing and Machine Learning for Electronic Trading and has been Associate Editor of IEEE TRANSACTIONS ON INFORMATION THEORY and of IEEE TRANSACTIONS ON SIGNAL PROCESSING, a Guest Editor of the IEEE SIGNAL PROCESSING MAGAZINE 2010 Special Issue on Convex Optimization for Signal Processing, the IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS 2008 Special Issue on Game Theory in Communication Systems, and the IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS 2007 Special Issue on Optimization of MIMO Transceivers for Realistic Communication Networks.