Capacity-Approaching *PhaseCode* for Low-Complexity Compressive Phase Retrieval

Ramtin Pedarsani, Kangwook Lee, and Kannan Ramchandran
Dept. of Electrical Engineering and Computer Sciences
University of California, Berkeley
{ramtin, kw1jjang, kannanr}@eecs.berkeley.edu

***Abstract*—In this paper, we tackle the general compressive phase retrieval problem. The problem is to recover (to within a global phase uncertainty) a K-sparse complex vector of length n, $x \in \mathbb{C}^n$, from the magnitudes of m linear measurements, $y = |Ax|$, where $A \in \mathbb{C}^{m\times n}$ can be designed, and the magnitudes are taken component-wise for vector $Ax \in \mathbb{C}^m$. We propose a variant of the PhaseCode algorithm, first introduced in [1], and show that under some mild assumptions, using an irregular left-degree sparse-graph code construction, the algorithm can recover almost all the K non-zero signal components using only slightly more than $4K$ measurements, with order-optimal time and memory complexity of $\mathcal{O}(K)$. It is known that the fundamental limit for the number of measurements in compressive phase retrieval problem is $4K - o(K)$ [2, 3]. To the best of our knowledge, this is the first constructive *capacity-approaching* compressive phase retrieval algorithm: in fact, our algorithm is also order-optimal in complexity and memory.**

I. Introduction

The general compressive phase retrieval problem is to recover a K-sparse signal of length n, $x \in \mathbb{C}^n$, to within a global phase uncertainty, from the magnitude of m linear measurements $y = |Ax|$, where $A \in \mathbb{C}^{m\times n}$ is the measurement matrix that can be designed, and the magnitude is taken on each component of the vector $Ax \in \mathbb{C}^m$.[1] In many applications such as optics, X-ray crystallography, astronomy, ptychography, quantum optics, etc., the phase of the measurements is not available. Furthermore, in many applications the signal of interest is sparse in some domain.

Main contribution. The main contribution of this paper is to propose a variant of PhaseCode algorithm, which was first introduced in [1]. The measurement matrix in PhaseCode algorithm is constructed based on a *sparse-graph code*. The work of [1], which features a regular left-degree distribution in its sparse-graph code design, establishes that, with high probability, all but a fraction 10^{-7} (error floor) of the non-zero signal components can be recovered using about $14K$ measurements with optimal time and memory complexity. In this work, we first consider an *irregular* left-degree distribution for the sparse-graph code construction. We show that under this construction, the PhaseCode algorithm can recover almost all the non-zero components of the signal using only slightly more than $4K$ measurements under some mild assumptions, with optimal time and memory complexity of $\mathcal{O}(K)$. It is well-known that $4K - o(K)$ measurements is the fundamental limit for unique recovery of a K-sparse signal x [2, 3]. This shows that the irregular PhaseCode algorithm is *capacity-approaching*. We note a few caveats, which are why we call our algorithm capacity-approaching. First, our irregular PhaseCode has an arbitrarily small error floor, which is however still non-vanishing as K goes to infinity. Secondly, in order to prove that our algorithm can exactly recover x, we assume that we start with a δ fraction of *known locations* of the active components of x, where δ is arbitrarily small but non-zero. This assumption is quite reasonable in practice, e.g. the low-frequency-band of most real-world signals of interest is typically non-zero. Another way to validate our assumption is through the use of a feedback stage in our design of so-called active measurement systems.

[1]This can be easily extended to the case where the signal x is sparse with respect to some other basis, for example in Fourier domain, by a simple transformation of the measurement matrix.

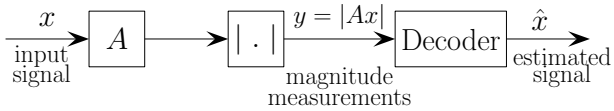


Fig. 1: Block diagram of the general compressive phase retrieval problem. The measurements are $y_i = |a_i x|$, where a_i is the i-th row of measurement matrix A.

Problem definition. Consider a K-sparse complex signal $x \in \mathbb{C}^n$ of length n. Define $A \in \mathbb{C}^{m\times n}$ to be the measurement matrix that needs to be designed. The phase retrieval problem is to recover the signal x from magnitude of linear measurements $y_i = |a_i x|$, where a_i is the i-th row of matrix A. Figure 1 illustrates the block diagram of our problem. The main objective of the general compressive phase retrieval problem is to design matrix A with low sample complexity (ideally close to $4K$) and a low-complexity decoding algorithm to reliably recover x.

Related work. To the best of our knowledge, the first algorithm for compressive phase retrieval was proposed by Moravec *et al.* in [4]. The "PhaseLift" method originally proposed for the non-sparse case [5], is also used for the sparse case in [6] and [7], requiring $\mathcal{O}(K^2\log(n))$ intensity measurements, and having a computational complexity of $\mathcal{O}(n^3)$, making the method less practical for large-scale applications. An alternating minimization method is proposed in [8] for the sparse case with $\mathcal{O}(K^2\log(n))$ measurements and a computation complexity of $\mathcal{O}(K^3 n\log(n))$. Compressive phase-retrieval via generalized approximate message passing (PR-GAMP) is proposed in [9], with good performance in both runtime and noise robustness shown via simulations. Jaganathan *et al.* consider the phase retrieval problem from Fourier measurements [10, 11]. They propose an SDP-based algorithm, and show that the signal can be provably recovered with $\mathcal{O}(K^2\log(n))$ Fourier measurements [10]. Cai *et al.* propose proposed SUPER algorithm for compressive phase-retrieval in [12]. The SUPER algorithm uses $\mathcal{O}(K)$ measurements and features $\mathcal{O}(K\log(K))$ complexity with zero error

floor asymptotically.

II. PhaseCode based on irregular left-degree sparse-graph codes

We consider the same architecture used in [1] to design our measurement matrix, A. We define $A \in \mathbb{C}^{pM \times n}$ to be a row tensor product of a trigonometric modulation matrix, $T \in \mathbb{C}^{p \times n}$, and a code matrix, $H \in \mathbb{C}^{M \times n}$. The row tensor product of matrices T and H is defined as follows. If $A = T \otimes H = [A_1^T, A_2^T, \ldots, A_M^T]^T$ and $A_i \in \mathbb{C}^{p \times n}$, then, $A_i(jk) = T_{jk} H_{ik},\ 1 \leq j \leq p,\ 1 \leq k \leq n$. As an example, the row tensor product of matrices

$$T = \begin{bmatrix} 0.1 & 0.2 & 0.3 \\ 0.4 & 0.5 & 0.6 \end{bmatrix} \text{ and } H = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

is

$$A = T \otimes H = \begin{bmatrix} 0 & 0.2 & 0 \\ 0 & 0.5 & 0 \\ 0.1 & 0.2 & 0 \\ 0.4 & 0.5 & 0 \\ 0 & 0 & 0.3 \\ 0 & 0 & 0.6 \end{bmatrix}.$$

Our trigonometric modulation matrix is the same as the one we proposed in [1]; that is,

$$T = \begin{pmatrix} e^{\mathbf{i}\omega} & e^{\mathbf{i}2\omega} & \ldots & e^{\mathbf{i}n\omega} \\ e^{-\mathbf{i}\omega} & e^{-\mathbf{i}2\omega} & \ldots & e^{-\mathbf{i}n\omega} \\ \cos(\omega) & \cos(2\omega) & \ldots & \cos(n\omega) \\ e^{\mathbf{i}\omega'} & e^{\mathbf{i}2\omega'} & \ldots & e^{\mathbf{i}n\omega'} \end{pmatrix}, \qquad (1)$$

where $\omega = \frac{2\pi}{n}$ and ω' is a random phase uniformly distributed in $[0, 2\pi)$.

Now we present the key idea of the PhaseCode algorithm.[2] For more details, we refer the readers to [1]. The architecture of the PhaseCode algorithm is as follows. We consider a bipartite graph of n left nodes and M right nodes, where each left node is a signal component and each right node is a set of 4 measurements. One can think of the n left nodes as n slots in which K balls will be thrown arbitrarily. The n slots refer to the indices of the n-length signal, and the K balls refer to the K non-zero components of the signal. See Figure 2 as an illustration. Our goal is to design the bipartite graph such that the K locations (belonging to the integer set between 1 and n) of these balls (non-zero components) are detected. Furthermore, the relative phase and magnitude of the non-zero components need to be detected with the aid of the trigonometric modulation matrix. (See Subsection II-A and [1] for details.)

In order to explain our design for the phase-retrieval problem, it is illustrative for us to consider an equivalent *ball-coloring* problem in the following sense. The goal of this equivalent problem is to detect and color all the balls in the system. We say that a ball is colored if it is detected that the ball corresponds to an active signal component in that slot, and the component is recovered in location index, magnitude and phase (upto a global phase). Thus, the ball-coloring problem is equivalent to the targeted phase-retrieval problem. However, the *rules* needed to color these balls are not yet clear, and need to be specified. This is where our carefully-designed trignometric-modulated measurements of Eq. (1) come in. Concretely, in our proposed construction, these trig-modulated measurements are used to enforce the following coloring rule. *If a right node of the bipartite graph is connected to $n' \geq 2$ balls and $n' - 1$ of those balls are colored, the single uncolored ball can get colored.* See Figure 3 as an illustration. What this means for our targeted phase-retrieval problem is that if we have already uncovered the locations and complex amplitudes (including phase) of $n' - 1$ of the n' signal components corresponding to a right node, then the trig-modulated measurements can be used to uncover the lone non-detected signal component; i.e., it too can be colored. Of course, at the outset of the algorithm, no balls are colored. Therefore, the algorithm requires an initial seeding phase which will jumpstart the process by ensuring that a few balls get colored. (We will show how this can be realized in Subsection II-B.) Given that after the initial phase of the algorithm, there are only a few colored balls in the graph, the goal is to design a bipartite graph such that the coloring spreads like an epidemic until all the balls get colored. Given this exact equivalence to our original phase-retrieval problem, in the interests of illustrative clarity, from now on, we mainly focus on the ball coloring problem, and connect it back to the phase retrieval problem in Subsection II-A.

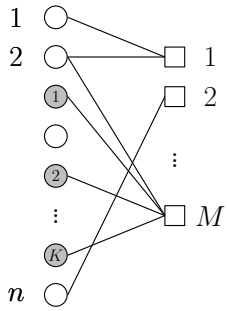


Fig. 2: Illustration of a bipartite graph representation of the compressive phase-retrieval problem. In this graph, each left node refers to a signal component and each right node refers to a set of (four) measurements. The signal is K-sparse, so K of these left nodes are active as illustrated in the figure. The four measurements associated with each right node correspond to the trigonometric-based modulation measurements of Equation (1).

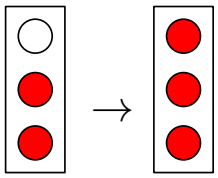


Fig. 3: This figure illustrates when a bin contains one or more colored balls and *exactly one* uncolored ball, then the uncolored ball gets colored.

As mentioned earlier, we design our code matrix based on a random bipartite graph model. Consider the pruned bipartite graph that is constructed by K left nodes or balls and M right nodes. From now on, we consider the pruned graph for analysis. Define ρ_i to be the probability that a randomly selected edge is connected to a right node of degree i, and λ_i to be the probability that a randomly selected edge is connected to a left node of degree i. Define the edge degree distributions or edge degree polynomials of right and left

[2] In this paper, we consider only the Unicolor PhaseCode algorithm proposed in [1].

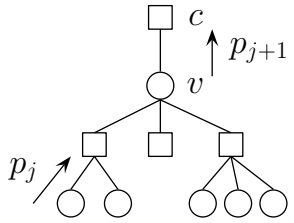


Fig. 4: Illustration of message passing algorithm and density evolution equation.

nodes as follows:

$$\rho(x) = \sum_{i\geq 1} \rho_i x^{i-1} \quad \text{and} \quad \lambda(x) = \sum_{i\geq 1} \lambda_i x^{i-1}.$$

To analyze the PhaseCode algorithm, we use *density evolution*, which is a technique to analyze the performance of message passing algorithms on sparse-graph codes. We find a recursion relating the probability that a randomly chosen ball or left node in the graph is not colored after j iterations of the algorithm, p_j, to the same probability after $j+1$ iterations of the algorithm, p_{j+1}. Our analysis methodology is similar to that of [1] with the distinction that we now consider a different left-degree distribution. We first use a tree-like assumption of the graph (i.e. the graph is free from cycles) and undertake a mean analysis using density evolution. Then, we relax the tree-like assumption, as well as find concentration bounds to show that the actual performance concentrates around the mean. Details follow. First, under the tree-like assumption of the graph, one has

$$p_{j+1} = \lambda(1 + \rho_1 - \rho(1 - p_j)). \tag{2}$$

Here is a proof of Equation (2). A right node (bin) passes a message to a left node (ball) that the left node can be colored if all of its neighbors are colored. This happens with probability $\sum_{i=2}^{\infty} \rho_i (1-p_j)^{i-1} = \rho(1-p_j) - \rho_1$. Thus, the probability that ball v passes a message to bin c that it is not colored in iteration $j+1$, is the probability that none of its other neighbor bins can tell v that it is colored in iteration j. See Figure 4 for an illustration. These messages are all independent if the bipartite graph is a tree. Assuming this, one has

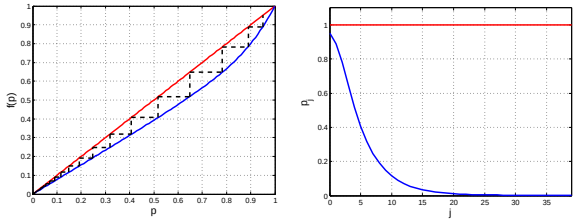
$$\begin{aligned} p_{j+1} &= \sum_{i\geq 1} \lambda_i [1 - (\rho(1-p_j) - \rho_1)]^{i-1} \\ &= \lambda(1 + \rho_1 - \rho(1-p_j)). \end{aligned} \tag{3}$$

Recall that our goal is to design a bipartite graph such that all the balls get colored with as few measurements as possible. The way that we design the bipartite graph is to connect the left nodes to right nodes uniformly at random according to some degree distribution. Then, for large K and M, the induced right node degree distribution is Poisson with parameter $\eta = \frac{K\bar{d}}{M}$, where $\bar{d}$ is the average degree of left nodes. Since ρ_i is the fraction of edges that are connected to a right node of degree i, we have

$$\rho_i = \frac{iM}{K\bar{d}} \mathbb{P}(\text{random right node has degree } i) = \frac{\eta^{i-1} e^{-\eta}}{(i-1)!}.$$

Then,

$$\rho(x) = \sum_{i\geq 1} \frac{\eta^{i-1} e^{-\eta}}{(i-1)!} x^{i-1} = e^{-\eta(1-x)}. \tag{4}$$



(a) (b)

Fig. 5: The density evolution curves for parameters $K = 10^5$, $\epsilon = 0.3$ and $D = 10^3$. Figure (a) illustrates the density evolution equation: $p_{j+1} = f(p_j)$. In order to track the evolution of p_j, pictorially, one draws a vertical line from (p_j, p_j) to $(p_j, f(p_j))$, and then a horizontal line between $(p_j, f(p_j))$ and $(f(p_j), f(p_j))$. Since the two curves meet at $(1,1)$ if $p_0 = 1$, then p_j gets stuck at 1. However, if $p_0 = 1-\delta$, for some $\delta > 0$, p_j decreases after each iteration, and it gets very close to 0 in finite number of iterations. Figure (b) illustrates the same phenomenon by showing the evolution of p_j versus the iteration, j. Note that in this example, since $\epsilon = 0.3$ and we are operating further away from the capacity, p_j gets very close to 0 after only 25 iterations.

We design the left degree distribution $\lambda(x)$ based on a truncated harmonic distribution, which is also known to be capacity-achieving in irregular LDPC codes for BEC channel [13]. Let $h(x) = \sum_{i=1}^{x} 1/i$. Then, $\lambda_i = \frac{1}{i-1} \times \frac{1}{h(D-1)}$, $2 \leq i \leq D$, where D is a (large) constant.

We design the number of right nodes to be $M = K/(1-\epsilon) \simeq K(1+\epsilon)$. How to choose constants D and ϵ will be shortly clarified in Lemma 1. The average degree of left nodes is $\bar{d} = \frac{1}{\sum_i \lambda_i/i}$ [13]. To see this, let E be the number of edges of the graph. Then, the number of left nodes of degree i is $E\lambda_i/i$ since λ_i is the fraction of edges with degree i on the left. Thus, the number of left nodes is $\sum_i E\lambda_i/i$. So the average left degree is $\bar{d} = \frac{E}{\sum_i E\lambda_i/i} = \frac{1}{\sum_i \lambda_i/i}$. Thus, with our design, $\bar{d} = h(D-1)\frac{D}{D-1}$. Consequently, the Poisson density parameter of the right-node degree distribution is

$$\eta = h(D-1)\frac{D}{D-1}(1-\epsilon).$$

Now we analyze the density evolution equation in (2). Let $f(x) = \lambda(1 + e^{-\eta} - e^{-\eta x})$. In Lemma 3.1. of [14], we show that given that $f'(1) > 1$, $f(x) = x$ has exactly two fixed points in the interval $[0,1]$. One fixed point is at 1, and the other fixed point of the equation is approximately $p^* \simeq \lambda(e^{-\eta})$, which corresponds to the error floor of the algorithm. Furthermore, $f(x) < x$ in the interval $(p^*, 1)$. See Figure 5a for an illustration. The evolution of p_j can then be described as follows. Assuming that one can get away from fixed point 1 to $1-\delta$ for some small $\delta > 0$, after a few iterations p_j gets very close to p^* as shown in Figures 5a and 5b. However, p_j cannot get smaller than p^*, so p^* is the error floor of the algorithm. In the following lemma, we show that for any arbitrarily small numbers p^* and ϵ, there exists a large enough constant $D(p^*, \epsilon)$ such that $f'(1) > 1$. This shows that with only $4M = 4K/(1-\epsilon) \simeq 4K(1+\epsilon)$ measurements, irregular PhaseCode algorithm can recover almost all the non-zero signal components. So given that the coloring procedure starts (the density evolution equation can be started from $1-\delta$), irregular PhaseCode is capacity-approaching. The proof of the following lemma is provided in [14] (Lemma 3.2).

Lemma 1. *For any $p^* > 0$ and any $\epsilon > 0$, there exists a large enough constant $D(\epsilon, p^*)$ such that $M = K(1-$*

$\epsilon)^{-1} \simeq K(1+\epsilon)$ *is the number of right nodes (bins), and* p_j *converges to* p^* *as* j *goes to infinity, given that* $p_1 = 1-\delta$. *Moreover, for any* $\epsilon_1 > 0$, *there exists a constant* $\ell(\epsilon_1)$ *such that* $p_\ell \leq p^* + \epsilon_1$.

In the density evolution analysis so far, we have shown that the *average* fraction of balls that remain uncolored will be arbitrarily close to the error floor after a fixed number of iterations, provided that the tree-like assumption is valid. It remains to show that the actual fraction of balls that are not in the giant component after ℓ iterations is highly concentrated around p_ℓ. Since the maximum degree of left nodes are again a constant D, the exact procedure used in [1] to get a concentration bound can be also applied here. Towards this end, first one can show that a neighborhood of depth ℓ of a typical edge is a tree with high probability for a constant ℓ.[3] Second, one uses the standard Doob's martingale argument as in [15], to show that the number of uncolored balls after ℓ iterations of the algorithm is highly concentrated around Kp_ℓ. We refer the readers to [1] for the proofs.

A. Trigonometric-Based Modulation

In this section, we will explain the choice of the trig-modulation T, and how T enables us to detect the non-zero signal components and recover them in magnitude and phase (up to a global phase). We will provide a brief explanation here, while referring the readers to [1] for thorough analysis.

Define the length 4 vector y_i to be the measurement vector corresponding to the i-th row of matrix H for $1 \leq i \leq M$. Then $y = [y_1^T, y_2^T, \ldots, y_M^T]^T$, where $y_i = [y_{i,1}, y_{i,2}, y_{i,3}, y_{i,4}]^T$. Let $\omega = \frac{\pi}{2n}$. Recall from Equation (1) that we design the measurement matrix $T = [t_{j\ell}]$ as follows: $t_{1\ell} = e^{\mathbf{i}\omega\ell}$, $t_{2\ell} = e^{-\mathbf{i}\omega\ell}$, $t_{3\ell} = 2\cos(\omega\ell)$, and $t_{4\ell} = e^{\mathbf{i}\omega'\ell}$. This measurement matrix enables the algorithm to detect a left node (bin) that is connected to some colored balls and only one uncolored ball. Furthermore, the measurements enable the algorithm to find the index of uncolored ball, and recover the corresponding non-zero signal component in magnitude and phase relative to the known (colored) components.

Consider a bin, let's say bin i, for which we know that it is connected to some colored balls. We want to check if bin i is connected to exactly one other uncolored ball; i.e. one unknown non-zero component of x, say x_ℓ, as in Figure 3. We now describe a guess and check strategy to check our guess, and to find ℓ and x_ℓ if the guess is correct. If our guess is correct, we have access to measurements of the form: $y_{i,1} = |a + e^{\mathbf{i}\omega\ell}x_\ell|$, $y_{i,2} = |b + e^{-\mathbf{i}\omega\ell}x_\ell|$, $y_{i,3} = |c + 2\cos(\omega\ell)x_\ell|$, and $y_{i,4} = |d + e^{\mathbf{i}\omega'\ell}x_\ell|$, where complex numbers a, b, c and d are known values that depend on the values and locations of the known colored balls. We want to solve the first 3 equations to find ℓ and x_ℓ, and use the last one ($y_{i,4}$) to check if our guess is correct. We skip the algebra here, and refer the readers to [1, 14] for details of how this can be done. We emphasize that one can solve the first 3 equations to find (at most) 4 possible solutions for x_ℓ and ℓ in closed-form. Whether one of those possible solutions is admissible or not can be checked using $y_{i,4}$. Since ω' is a random phase, the probability that the check equation declares admissible solution while the solution is not valid is 0.

B. How to initialize the ball-coloring algorithm?

We have so far assumed that an arbitrarily small fraction $\delta > 0$ of the balls are colored in the initial phase. That is, a positive fraction of non-zero signal components are detected. In this subsection, we propose two methods for initializing the coloring algorithm. The first method is based on having an *active* sensing stage. In active sensing, the measurements can be adaptive and are functions of the previous observations. The second method assumes that the *locations*[4] of a small fraction of non-zero components of the signal are known. Note that this is the case for most practical scenarios. For example, almost all sparse images in the Fourier domain have non-zero low band components. In both methods, we make essential use of the deterministic measurements for non-sparse phase retrieval introduced in [1].

In the first method, as mentioned, we first try to find the location of a positive fraction of non-zero components. Consider the following $3\epsilon_2 K$ measurements[5] for some small $\epsilon_2 > 0$:

$$\tilde{y} = |T_1 \otimes H_1 x|,$$

where

$$T_1 = \begin{pmatrix} 1 & 1 & \ldots & 1 \\ e^{\mathbf{i}\omega} & e^{\mathbf{i}2\omega} & \ldots & e^{\mathbf{i}n\omega} \\ \cos(\omega) & \cos(2\omega) & \ldots & \cos(n\omega) \end{pmatrix} \in \mathbb{C}^{3\times n}, \quad (5)$$

and $H_1 \in \{0,1\}^{\epsilon_2 K \times n}$ is a code matrix, and is constructed as follows. Each column of H_1 has exactly one entry that is equal to 1, and the location of this entry is random. Equivalently, we consider a random bipartite graph model in which left nodes (which refer to signal components) are columns of H_1 and right nodes (which refer to measurements) are rows of H_1. Let $\tilde{y} = [\tilde{y}_1^T, \tilde{y}_2^T, \ldots, \tilde{y}_{M_1}^T]^T$, where $\tilde{y}_i = [\tilde{y}_{i,1}, \tilde{y}_{i,2}, \tilde{y}_{i,3}]^T$. Define a *singleton* right node to be a right node that is connected to exactly 1 left nodes with non-zero signal component. Now we show that if a right node is a singleton, the location and magnitude of the non-zero component can be found using ratio test. To this end, suppose that $\tilde{y}_{i,1} = |x_\ell|$, $\tilde{y}_{i,2} = |e^{\mathbf{i}\ell\omega}x_\ell|$, and $\tilde{y}_{i,3} = |\cos(\ell\omega)x_\ell|$. The fact that $\frac{\tilde{y}_{i,1}}{\tilde{y}_{i,2}} = 1$ reveals that the i-th right node is a singleton. The magnitude of the non-zero component is $\tilde{y}_{i,1}$. Furthermore, the location of the non-zero component can be found using $\ell = \cos^{-1}(\tilde{y}_{i,3}/x_\ell)$. It is easy to see that the number of active components that can be found in location and magnitude approaches $K(1-\frac{1}{\epsilon_2 K})^K = K\delta$, where $\delta = e^{-1/\epsilon_2} > 0$ is a constant. In order to recover the phases of these non-zero components, we use deterministic measurements as follows. Let $\delta K = K_1$. Without loss of generality and for ease of notation assume that the detected non-zero components are $x_1, x_2, \ldots, x_{K_1}$. We use $2K_1 - 2$ measurements to get $|x_1 + x_\ell|$ and $|x_1 + \mathbf{i}x_\ell|$ for $2 \leq \ell \leq K_1 - 1$. Then, one can easily find the relative angle between x_ℓ and x_1 similar to the procedure described

[3] The depth-ℓ neighborhood of edge (v, c) is the subgraph of all the edges and nodes of paths having length less than or equal to ℓ, that start from v and the first edge of the path is not (v, c).

[4] Note that we assume only *location* knowledge of a tiny fraction of the active signal components, but no knowledge about the magnitude or phase of these components.

[5] Without loss of generality suppose that $\epsilon_2 K$ is an integer.

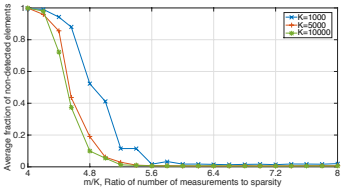


Fig. 6: Performance of irregular PhaseCode *without initialization phase*. We observe that even without the initialization stage, the coloring algorithm successfully recovers the signal with a sample complexity that is close to the fundamental limit $4K$.

in Section 4.1. of [1]. Therefore, a positive fraction δ of the active signal components can be found in location, phase and magnitude, and their corresponding balls can be colored. This ensures that the ball-coloring algorithm is initialized and it recovers almost all the signal components. Moreover, the number of extra measurements that we use in the active sensing stage is only $(3\epsilon_2 + 2\delta)K$, and $(3\epsilon_2 + 2\delta) > 0$ is a constant which can be made arbitrarily small. Thus, the total number of measurements that PhaseCode uses is still $4(1+\epsilon)K$ for arbitrarily small constant $\epsilon > 0$.

The second method assumes that the location of a small fraction δ of active signal components is known, e.g. the low band components of an image in Fourier domain. Given this, one can use exactly the same procedure described in the first method to recover these components with $3\delta K - 2$ measurements.

III. Conclusion

In this paper, we have considered the problem of recovering a K-sparse complex signal $x \in \mathbb{C}^n$ from m intensity measurements of the form $|a_i x|$, $1 \leq i \leq m$, where a_i is a measurement row vector. We have proposed an irregular-left-degree PhaseCode algorithm that can recover the signal almost perfectly using only $4K(1+\epsilon)$ measurements for arbitrarily small ϵ with high probability as K gets large under some mild assumptions.

We conclude the paper by emphasizing an interesting empirical observation. In Section II-B, we explained how one can initialize the coloring algorithm by means of feedback or prior knowledge about a few non-zero components of the signal. In our previous work [1], we showed that a linear-size component of colored balls can be formed with around $14K$ measurements (which is far from the fundamental limit $4K$) in the regular construction without using feedback or any prior knowledge about the signal. However, via simulations, we empirically observe that our coloring algorithm can be used even without the initialization stage for irregular PhaseCode. We simulate irregular PhaseCode without any initialization stage with the following parameters. We vary K from 1000 to 10000, and fix the length of signal as $n = 20000$. We observe that even without the initialization stage, the coloring algorithm successfully recovers the signal with a sample complexity that is close to the fundamental limit. For example, when $K = 10000$, the coloring algorithm successfully decodes the signal with only $5.2K$ measurements. An interesting future direction is to theoretically investigate whether irregular PhaseCode without initialization is also capacity-approaching. Finally, in this work, we have focused on the noiseless compressive phase-retrieval problem, predominantly to elucidate our sparse-graph based capacity-approaching construction. Of course, in most practical settings, one needs to consider noisy observations. Our proposed framework can indeed be extended to be robust to noise [16].

References

[1] R. Pedarsani, K. Lee, and K. Ramchandran, "Phasecode: Fast and efficient compressive phase retrieval based on sparse-graph-codes," *arXiv preprint arXiv:1408.0034*, 2014.

[2] M. Akcakaya and V. Tarokh, "New conditions for sparse phase retrieval," *arXiv preprint arXiv:1310.1351*, 2013.

[3] T. Heinosaari, L. Mazzarella, and M. M. Wolf, "Quantum tomography under prior information," *Communication in Mathematical Physics*, vol. 318, no. 2, pp. 355–374, 2013.

[4] M. L. Moravec, J. K. Romberg, and R. Baraniuk, "Compressive phase retrieval," *SPIE Conf. Series*, vol. 6701, 2007.

[5] E. J. Candes, T. Strohmer, and V. Voroninski, "Phaselift: Exact and stable signal recovery from magnitude measurements via convex programming," *Communications on Pure and Applied Mathematics*, vol. 66, no. 8, pp. 1241–1274, 2013.

[6] H. Ohlsson, A. Yang, R. Dong, and S. Sastry, "Compressive phase retrieval from squared output mea- surements via semidefinite programming," *arXiv preprint arXiv:1111.6323*, 2011.

[7] X. Li and V. Voroninski, "Sparse signal recovery from quadratic measurements via convex programming," *arXiv preprints arXiv:1209.4785*, 2012.

[8] P. Netrapalli, P. Jain, and S. Sanghavi, "Phase retrieval using alternating minimization," *arXiv preprints arXiv:1306.0160*, 2013.

[9] P. Schniter and S. Rangan, "Compressive phase retrieval via generalized approximate message passing," in *Proceedings of Allerton Conference on Communication, Control, and Computing*, 2012.

[10] K. Jaganathan, S. Oymak, and B. Hassibi, "Sparse phase retrieval: Convex algorithms and limitations," pp. 1022–1026, 2013.

[11] ——, "Phase retrieval for sparse signals using rank minimization," in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*, 2012, pp. 3449–3452.

[12] S. Cai, M. Bakshi, S. Jaggi, and M. Chen, "Super: Sparse signals with unknown phases efficiently recovered," *arXiv preprint arXiv:1401.4451*, 2014.

[13] M. Luby, M. Mitzenmacher, M. A. Shokrollahi, and D. Spielman, "Improved low-density parity check codes using irregular graphs," *IEEE Trans. Info. Theory*, vol. 47, pp. 585–598, 2001.

[14] R. Pedarsani, K. Lee, and K. Ramchandran, "Capacity-approaching phasecode for low-complexity compressive phase retrieval," *arXiv preprint arXiv:1412.5694*, 2014.

[15] T. Richardson and R. Urbanke, "The capacity of low-density parity-check codes under message-passing decoding," *IEEE Transactions on Information Theory*, vol. 47, pp. 599–618, February 2001.

[16] D. Yin, K. Lee, R. Pedarsani, and K. Ramchandran, "Fast and robust compressive phase retrieval with sparse-graph codes," *submitted to ISIT*, 2015.