PhaseCode: Fast and Efficient Compressive Phase Retrieval based on Sparse-Graph-Codes

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

We consider the problem of recovering a 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 address multiple settings corresponding to whether the measurement vectors are unconstrained choices or not, and to whether the signal to be recovered is sparse or not. However, our main focus is on the case where the measurement vectors are unconstrained, and where x is exactly K-sparse, or the so-called general compressive phase-retrieval problem.

We introduce ***PhaseCode***, a novel family of fast and efficient merge-and-color algorithms (that includes ***Unicolor PhaseCode*** and ***Multicolor PhaseCode***) that are based on a sparse-graph-codes framework. As one instance, our *Unicolor PhaseCode* algorithm can provably recover, with high probability, all but a tiny 10^{-7} fraction of the significant signal components, using at most $m = 14K$ measurements, which is a small constant factor from the fundamental limit, with an optimal $\mathcal{O}(K)$ decoding time and an optimal $\mathcal{O}(K)$ memory complexity. Next, motivated by some important practical classes of optical systems, we consider a "Fourier-friendly" constrained measurement setting, and show that its performance matches that of the unconstrained setting. In the Fourier-friendly setting that we consider, the measurement matrix is constrained to be a cascade of Fourier matrices (corresponding to optical lenses) and diagonal matrices (corresponding to diffraction mask patterns). We also study the general non-sparse signal case, for which we propose a simple deterministic set of $3n - 2$ measurements that can recover the n-length signal under some mild assumptions. Throughout, we provide extensive simulation results that validate the practical power of our proposed algorithms for the sparse unconstrained and Fourier-friendly measurement settings, for noiseless and noisy scenarios. A key contribution of our work is the novel use of coding-theoretic tools like density evolution methods for the design and analysis of fast and efficient algorithms for compressive phase-retrieval problems. This contrasts and complements popular approaches to the phase retrieval problem based on alternating-minimization, convex-relaxation, and semi-definite programming.

1 Introduction

1.1 Phase Retrieval Problem

Compressive sensing (CS) has recently emerged as a powerful framework for understanding the fundamental limits for signal acquisition and recovery [18, 24]. The basic premise of CS is that a high-dimensional signal that is sparse in some basis, can be recovered from linear projections of the signal with respect to an appropriate lower-dimensional measurement system. A key attribute of CS is that the measurement system is linear and phase-preserving. That is, the acquired samples, complex-valued in general, contain both the magnitude and phase of the measurements.

In many applications of interest, e.g. related to optics [28], X-ray crystallography [8, 9], astronomy [12], ptychography [20], quantum optics [29], etc., the phase information in the measured samples is not available. For example, in optical systems, one can measure only the intensity of the measurements as they relate to the photon count on a detector. Thus, the phase of the measurements is lost. Indeed, the problem of recovering a signal from only the magnitude of its Fourier transform has been a well-studied problem in the signal processing literature for several decades under the umbrella of phase-retrieval [10]. It has recently received renewed interest in the "post-compressed-sensing" era [14, 19, 23], allowing for the insights from compressive sensing to be incorporated into the phase-retrieval problem when the signal of interest is sparse, and the measurement matrix is unconstrained.

Concretely, consider a signal $x \in \mathbb{C}^n$ and a measurement matrix $A \in \mathbb{C}^{m \times n}$. The phase-retrieval problem is to recover x from the observations $y = |Ax|, x \in \mathbb{C}^n$, where the magnitude is taken on each element of the vector Ax. The compressive phase retrieval problem targets the case where x is K-sparse.

In this paper, we study the phase-retrieval problem under the following settings:

(i) General compressive phase-retrieval of sparse signals [1];

(ii) "Fourier-friendly" compressive phase-retrieval of signals having a sparse spectrum; and

(iii) Phase-retrieval of signals that are not sparse, under both the general and Fourier-friendly measurement settings.

We now summarize each of these settings:

(i) **General compressive phase-retrieval of sparse signals**: In this setting, which we discuss in detail in Section 2, we are free to design the measurement matrix A without any constraints, and this represents the primary contribution of this paper. We consider it for three reasons.

(1) It is of broadest theoretical interest, being the most general compressive-phase-retrieval problem, for which we propose a sparse-graph-codes framework that is a significant departure from currently popular approaches based on convex relaxation, Semi-Definite Programming (SDP), alternating minimization, etc. [15, 21–23, 25, 26].

(2) It provides the intellectual insights and the foundational framework needed to address more constrained problems, such as those studied under the Fourier-friendly setting of category **(ii)**.

(3) It is of independent interest in applications related to certain quantum optical systems. For example, compressive sensing has been used in recent work involving quantum optics [29] to measure the transverse wavefunction of a photon, where the design of the measurement matrix has no constraints.

(ii) **Fourier-friendly compressive phase-retrieval of signals having a sparse spectrum**: In this category, motivated by applications related to Fourier optical systems, the measurement matrix A is constrained to be Fourier-friendly (see Section 3 for a detailed treatment). Concretely, A is constrained to be the cascade of (up to a couple of) stages of a diagonal matrix (corresponding to a so-called optical mask or coded diffraction pattern) and a Fourier transform (corresponding to an optical lens). This constraint is motivated by practical optical systems [30], array imaging [11], etc., as also addressed recently by [17].

(iii) **Phase-retrieval of general non-sparse signals**: Finally, in the interests of completeness, we address the case where the signal of interest is not sparse; i.e., the classical phase-retrieval problem. Under this category, we address both the general (unconstrained A) as well as the Fourier-friendly (constrained A) settings. See Section 4 for details.

[1] This is easily extended, as is well known, to the case where the signal x is sparse w.r.t. some other basis, such as a wavelet, but in the interests of conceptual clarity, we will not consider such extensions in this work.

The phase-retrieval problem has been studied extensively over several decades. We do not attempt to provide a comprehensive literature review here; instead, we highlight here only some of the pertinent and diverse approaches to this problem that we are aware of. A large body of literature is dedicated to the phase-retrieval problem for the case where the signal to be recovered has no structure and is non-sparse. "Phaselift" proposed by Candes *et al.* [15] and "PhaseCut" proposed by Waldspurger *et al.* [33] are examples of convex relaxation methods to solve the problem using semi-definite programming using $\mathcal{O}(n\log(n))$ measurements. While algorithms based on SDP provide theoretical performance guarantees and are robust to noise, they suffer from a high computational complexity of $\mathcal{O}(n^3)$ rendering them unsuited for many practical applications that require n to scale[2]. In [26], the authors propose an efficient algorithm based on alternating minimization that reconstructs the signal with $\mathcal{O}(n\log(n)^3)$ measurements.

In [4,7,13,31], several sets of authors investigate the fundamental limits of phase retrieval problem, with the goal of finding necessary or sufficient conditions on the minimum number of measurements needed to guarantee that the solution is unique. In summary, $4n-4$ measurements are shown to be sufficient [31], and $4n-o(n)$ measurements are necessary [13] to reconstruct any signal perfectly.

We now review some relevant literature on compressive phase retrieval. To the best of our knowledge, the first algorithm for compressive phase retrieval was proposed by Moravec *et al.* in [14]. This approach requires knowledge of the ℓ_1 norm of the signal, making it impractical in most scenarios. The authors in [32] showed that $4K-1$ measurements are theoretically sufficient to reconstruct the signal, but did not propose any algorithm. The "PhaseLift" method is also proposed for the sparse case in [23] and [25], 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. The alternating minimization method in [26] can also be adapted to the sparse case with $\mathcal{O}(K^2\log(n))$ measurements and a complexity of $\mathcal{O}(K^3n\log(n))$. Compressive phase-retrieval via generalized approximate message passing (PR-GAMP) is proposed in [19], with good performance in both runtime and noise robustness shown via simulations without theoretical proofs.

A common attribute of all of the above-mentioned compressive phase retrieval references is that they assume that the measurement matrix can be designed arbitrarily. This renders them inapplicable to many practical constrained settings such as Fourier-optical systems. In [17], Candes *et al.* consider measurement matrices that are Fourier-friendly as described in the previous subsection, but only for the non-sparse case. They show that "PhaseLift" is able to recover the signal with $\mathcal{O}(n\log(n)^4)$ measurements by using $\mathcal{O}(\log(n)^4)$ masks or coded diffraction patterns. For the sparse case, Jaganathan *et al.* consider the phase retrieval problem from Fourier measurements only [21, 22],. They propose an SDP-based algorithm, and show that the signal can be provably recovered with $\mathcal{O}(K^2\log(n))$ Fourier measurements [21]. They also propose a combinatorial algorithm for the case that the measurement matrix can be designed without constraints, and show that the signal can be recovered with $\mathcal{O}(K\log(n))$ measurements and time complexity of $\mathcal{O}(Kn\log(n))$ [21].

In the prior literature that we are aware of, the works which overlap most in spirit with ours are (i) the recently proposed SUPER algorithm for compressive phase-retrieval by Cai *et al.* in [2]; and (ii) the FFAST algorithm of Pawar and Ramchandran [6] which also features the use of coding-theoretic tools for efficiently computing a sparse Discrete Fourier Transform. With regard to the FFAST algorithm [6], despite the common use of coding-theoretic tools, our problem formulation, analysis, and resulting algorithm are really significantly different, mainly because our problem involves the loss of measurement phase, unlike that of FFAST. (See Section 2 for details.)

With regard to the SUPER algorithm of [2], again, while there are some similarities between the two approaches – mainly to do with the use of certain system subcomponents such as similar (but not identical)

[2]This limits the use of SDP-based methods to small to moderate values of n in practice. In contrast, we show simulations in the paper where n can be very large, even as large as 10^{10}. See Figures 6 and 7

trigonometric-modulation method to resolve phase ambiguities, and the common use of a giant-component-cluster in the initial phase of our proposed Unicolor PhaseCode algorithm (see Section 2.3 for details), our works are significantly distinct at many levels. First, the SUPER algorithm targets only the general unconstrained compressive phase-retrieval setting, whereas, as described earlier, we also target Fourier-friendly constrained settings that are applicable in practical optical systems, as well as non-sparse signal settings for both general and Fourier-friendly measurement systems. Secondly, even in the unconstrained phase-retrieval setting, there are significant distinctions between the two works with respect to theory, algorithm, and performance guarantees. As a quick overview, the SUPER algorithm uses $\mathcal{O}(K)$ measurements and features $\mathcal{O}(K\log(K))$ complexity with a zero-error-floor asymptotically. In contrast, by trading off the zero-error-floor for an arbitrarily-small controllable error-floor, our solution features key advantages. Specifically, this allows us to design more efficient measurement systems that are based on a new and novel sparse-graph-codes framework, and to characterize the precise number of measurements needed (featuring provably small constants that are a small factor from the fundamental limit, rather than only Big Oh statements), and, most importantly, this permits us to feature an optimal $\mathcal{O}(K)$ decoding algorithm with optimal $\mathcal{O}(K)$ memory requirements. Finally, we also demonstrate how our proposed solution is robust to noise with some modifications, and provide extensive validating simulation results.

1.2 Main Contributions

As mentioned earlier, the key contribution of this work is in the introduction of coding theory techniques such as density evolution and sparse-graph-codes to lay the theoretical and algorithmic foundations for the general compressive phase-retrieval problem. This allows us to come up with a provably efficient and fast PhaseCode family of algorithms that are order-optimal in terms of number of measurements needed, time-complexity, and memory-complexity, which are all $\mathcal{O}(K)$. Furthermore, we provide precise constants for the number of measurements needed to achieve a targeted reliability as defined in Section 2. To the best of our knowledge, this is the first work that provides precise constants for the number of measurements that are a small factor from the fundamental lower bound. As a specific operating point, our proposed Unicolor PhaseCode algorithm can provably recover a fraction of at least $1-10^{-7}$ of the active signal components with $14K$ measurement, with an asymptotically high reliability of $1-\mathcal{O}(1/K)$. This is one instance of an entire family of trade-offs between the number of measurements needed and the fraction of non-zero signal components that can be recovered using PhaseCode.

Another key contribution of this work is to adapt the PhaseCode algorithm to a more constrained Fourier-friendly setting that is useful in certain optical systems. Specifically, we show how it is possible to elegantly integrate the Chinese-Remainder-Theorem-centric framework of Pawar and Ramchandran [6] (that was used to find a fast sparse Discrete-Fourier-Transform) into our PhaseCode framework without any loss of system performance in terms of measurement cost or computational complexity. See Section 3 for details.

Next, we address the non-sparse case, and propose a set of $3n-2$ measurements that guarantee unique reconstruction of the signal under some mild assumptions.[3] This set of measurements can also be achieved using only 3 diagonal matrices (diffraction masks) and Fourier blocks (optical lenses). See Section 4 for details.

We provide pseudocode (in Appendix C) and an extensive set of simulation results for all of the above settings that validate our theoretical findings, and verify the close match between theory and practice. In this regard, we go beyond the UniColor PhaseCode algorithm, which comes with strong theoretical guarantees, to the MultiColor PhaseCode algorithm, which outperforms the UniColor PhaseCode algorithm empirically, but whose theoretical analysis remains open, and will be part of our ongoing and future work. Concretely, for the same instance cited earlier for the UniColor PhaseCode operating point (recovery of $1-10^{-7}$

[3] These mild assumptions ensure that there is no contradiction between our results and fundamental limits based on *injectivity* requirements studied in the literature [7].

fraction of the active signal components with $14K$ measurements), the more efficient MultiColor PhaseCode algorithm is shown to need only about $11K$ measurements. Simulations confirm the runtime and memory requirements of our proposed algorithms as being linear in K and independent of n. This allows us to run PhaseCode using parameters as high as $n = 10^{10}$ and $K = 10^4$ on a regular laptop. See Figure 7.

Finally, our baseline PhaseCode algorithm can be modified in a modular fashion to be robust to noise. We present simulation results verifying this at the end of this paper, with of course commensurately increased cost in terms of both number of measurements needed for noise robustness. In the interests of conceptual clarity of the new ideas and tools that we bring in this work, we do not undertake a robustness analysis of PhaseCode, leaving that to ongoing and future work. However, we do want to emphasize here that the underlying architecture of our proposed PhaseCode algorithm is based on a kind of "separation principle" that admits a modular approach to achieving robustness. Specifically, the core sparse-graph-codes framework remains the same, with the same underlying merge-and-color philosophy as in the noiseless setting. What changes is the "trigonometric-modulation" measurement subsystem which needs to be appropriately robustified to deal with noise. Our modular architecture allows us to address both noiseless and noisy settings very efficiently. Specifically, it allows us to maximally leverage the K-sparse signal structure in the noiseless case by requiring a measurement cost and runtime complexity that are order-optimal $\mathcal{O}(K)$ *with no dependency on the ambient signal dimension* n. This is in contrast with most existing approaches to the compressive phase-retrieval problem that are based on SDP, convex relaxation, GAMP, etc. [15, 19] whose measurement cost and complexity depend on n even in the noiseless scenario. We believe that this is a key intellectual distinction of our approach, which can be systematically modified in a modular fashion to be robust to noise, and will be part of our future publication on this topic.

Table 1 summarizes our contributions.

1.3 Paper Organization

The rest of the paper is organized as follows. In Section 2, we consider the general compressive phase retrieval where the signal x is K-sparse. The Unicolor PhaseCode and Multicolor PhaseCode algorithms to recover x, are proposed in Subsection 2.3. The main theorem of the paper is also provided in this subsection. The analysis of Unicolor PhaseCode and the proof of the main theorem is provided in Subsection 2.4. Via extensive simulations, we evaluate both Unicolor and Multicolor PhaseCode algorithms, validating the theorem. In Section 3, we demonstrate how our proposed measurements for the sparse case can be obtained in a Fourier-friendly setting. In Section 4, we consider the case that x is non-sparse, and provide a simple yet effective set of measurements to recover the signal, in both general and Fourier-friendly settings. Finally, the paper is concluded in Section 5.

2 Sparse Case

2.1 Problem Formulation

Consider a complex signal $x \in \mathbb{C}^n$ of length n which is exactly K-sparse; that is, only K out of n components of vector x are non-zero. Let $A \in \mathbb{C}^{m \times n}$ be the measurement matrix that needs to be designed. The phase retrieval problem is to recover the signal x from magnitude 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 objectives of the general compressive phase retrieval problem are to design matrix A, and the decoding algorithm to recover x, that satisfy the following objectives.

- The number of measurements m is as small as possible. Ideally, one wants m to be close to the fundamental limit of $4K - \mathcal{O}(1)$ [32].

- The decoding algorithm is fast with low computational complexity and memory requirements. Ideally,

	General	Fourier-friendly
K-sparse	*Multicolor and Unicolor PhaseCode Algorithms* • Section 2 (proofs and simulations), Appendix C (pseudocode). • Number of measurements is $\mathcal{O}(K)$ with provably small constants, e.g., at least $1-10^{-7}$ fraction of the active signal components is recoverable with $14K$ measurements, w.h.p. • Time & memory complexity are order-optimal, $\mathcal{O}(K)$.	Fourier-constrained *PhaseCode Algorithms* • Section 3 (proofs and simulations) • A constrained version of *PhaseCode* useful for optical systems consisting of only diagonal diffraction masks and optical lenses. • Number of measurements and time & memory complexity are the same as in the general PhaseCode algorithms.
Non-sparse	General Phase Retrieval algorithm • Section 4 • We show that $3n-2$ deterministic measurements suffice to uniquely recover x under mild assumptions.	Fourier-constrained General Phase Retrieval algorithm • Section 4.1 • We show that $3n$ deterministic measurements suffice to uniquely recover x under mild assumptions. • Recovery of x is possible with only 3 uses of one mask and one lens.

Table 1: Summary of the main contributions of the paper.

one wants the time complexity and the memory complexity of the algorithm to be $\mathcal{O}(K)$, which is optimal.

- The reliability of the recovery algorithm should be maximized. Ideally, one wants the probability of failure to be vanishing as the problem parameters K and m get large.[4]

2.2 Main Idea of the PhaseCode Algorithm for Compressive Phase Retrieval

In this section, we briefly describe the main idea of PhaseCode. Decoding a message from a received signal of encoded symbols, where some of these symbols are subjected to erasure or corruption by a communication channel, has been studied extensively in Coding theory [38]. The compressive phase retrieval problem has some similarities to decoding over packet-erasure channels in the sense that $n-K$ symbols or signal

[4]In this work, we are interested in the asymptotic K regime. However, even when K is small, with proper modification of our algorithm, high reliability can be guaranteed when m gets large. We do not discuss this any further in the interest of presentation clarity.

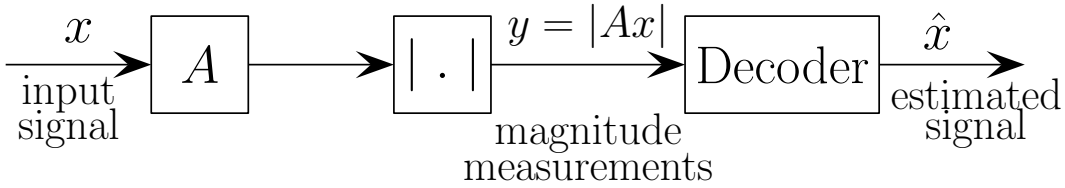


Figure 1: Block diagram of general compressive phase-retrieval problem. The measurements are $y_i = |a_i x|$, where a_i is the i-th row of measurement matrix A. The objectives are to design measurement matrix A and the decoding algorithm to guarantee high reliability, while having small sample complexity as well as small time and memory complexity.

components are known to be 0 while K of them are unknown or erased. Of course, unlike as in erasure channels, where the identity of erased symbols is known to the decoder, in our problem we do not know which signal components are non-zero, complicating life much more. Another major difference is the fact that the phase information of the measurements is not available in our problem. Erasure codes designed on sparse graphs such as [34, 35] are known to have fast iterative peeling-based decoders, and they are almost capacity-achieving. These attractive properties of sparse-graph codes in terms of both performance and complexity, inspire us to avail of the rich toolkit of coding theory to tackle the compressive phase retrieval problem.

Before we consider the compressive phase-retrieval problem, let us first provide a brief illustrative overview of the peeling-decoder, which is popular in coding applications. Consider the following simple example of solving a system of equations to solve for 4 unknown variables $x_1, \ldots, x_4$ from 4 linear equations:

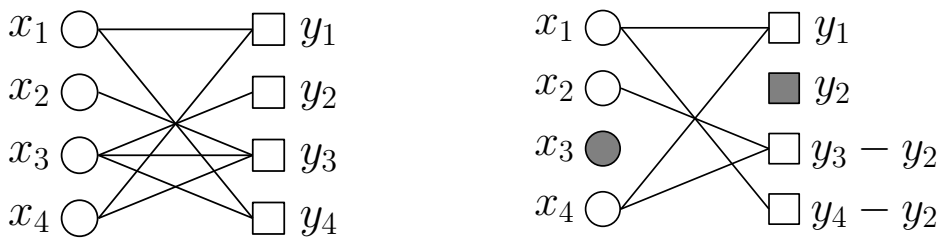
$$y_1 = x_1 + x_4; \tag{1}$$
$$y_2 = x_3; \tag{2}$$
$$y_3 = x_2 + x_3 + x_4; \tag{3}$$
$$y_4 = x_1 + x_3. \tag{4}$$

These equations can also be conveniently represented using a bipartite graph, or using a balls-and-bins model. In this representation, left nodes are variables or balls, and right nodes are measurement equations or bins. If left node i is connected to right node j, we say that ball i is in bin j. The graph of Figure 2a is a bipartite graph representation of Equations (1)–(4). Note that this example is much simpler than a compressive phase-retrieval problem because there is no ambiguity about the locations of a sparse set of non-zero variables, and furthermore, the phase information is known.

As in [6], we use the following terminology extensively throughout the paper:

- *Singleton:* A measurement equation is a singleton if it involves only one variable. Equation (2) is an example of a singleton equation. Equivalently, we define a right node of the bipartite graph or a bin to be a singleton if it has degree one.

- *Doubleton:* A measurement equation is a doubleton if it involves two variables. Equation (1) is an example of a doubleton equation. Equivalently, we define a right node of the bipartite graph or a bin to be a doubleton if it has degree two.



(a) Bipartite graph of Equations (1)–(4)

(b) Bipartite graph of Equations (5)–(7) after peeling x_3

Figure 2: Bipartite graph (Balls and bins) representation. The left figure represents the system of equations (1)–(4). In this graph, the bin corresponding to y_2 is a singleton, so x_3 can be uncovered and peeled off from other equations. Thus, the edges connected to the ball corresponding to x_3 are peeled off after x_3 is uncovered. The subsequent graph is shown in the right figure.

- *Multiton:* A measurement equation is a multiton if it involves more than one variable.[5] Equations (3) and (1) are examples of multiton equations. Equivalently, we define a right node of the bipartite graph or a bin to be a multiton if it has degree larger than one.

A peeling decoder works by "peeling off" variables or balls from singleton bins or singleton equations. In our toy example, x_3 is in a singleton bin, y_2, and can be peeled from the other equations to get

$$y_1 = x_1 + x_4 \tag{5}$$
$$y_3 - y_2 = x_2 + x_4 \tag{6}$$
$$y_4 - y_2 = x_1. \tag{7}$$

As a result of the peeling operation at step 1, x_1 can now be recovered from (7). In this way, the decoder keeps recovering the singletons to uncover all the variables. The bipartite graph of Figure 2b is a representation of Equations (5)–(7) after peeling x_3. To summarize, at each iteration of the peeling algorithm, the decoder uncovers the "new" singleton bins after the edges corresponding to uncovered variables are peeled off.

The balls-and-bins or bipartite-graph of Figure 2a is also a convenient way to represent our problem where the left nodes (balls) represent the non-zero signal components, and the right nodes (bins) represent the (magnitude of the) measurements. However, we do not readily know which components of the signal are non-zero. For now we assume that a genie will provide us with this location information. We show in Sections 2.3 and 2.5 how this genie can be realized using modulated measurements based on trigonometry. Another key complication in our problem is the lack of phase knowledge in our magnitude-only measurement system. This is a crucial drawback, as it renders classical singleton-based peeling and component recovery impossible, as phase information is critical to doing successful peeling.

In Section 2, we show how to circumvent this obstacle by using geometry, or more precisely, a carefully designed small set of trigonometric measurements to "modulate" a baseline sparse-graph code. While we refer the reader to Section 2 for the technical details, we try to provide some high-level intuition here. As is well-known and also intuitive, in the phase-retrieval problem, the signal of interest can be recovered only to within an unknown global phase. The idea is to start with an arbitrarily chosen single component, give it global zero-phase, and align all other recovered signal components with respect to it. This suggests the intuition of building up one or more giant clusters of components (balls), where in our terminology, these clusters are identified by their colors; i.e. all the balls belonging to a particular cluster have the same

[5] In our terminology, a doubleton bin is also a multiton bin.

color. Two (or more) balls can be colored with the same color if they represent signal components whose phases are aligned with respect to each other. The trigonometric-modulated measurements empower our PhaseCode algorithms with the ability to align component phases (or in more colorful terminology, to color balls) using plane-geometric arguments (such as using the cosine-law to find the relative phase between two components x and y, given $|x|$, $|y|$, and $|x+y|$).

So, in summary of the big picture, our proposed PhaseCode family of solutions features coloring-based algorithms, corresponding to having a single giant cluster (Unicolor) or several clusters (Multicolor). As mentioned earlier, we can rigorously analyze the Unicolor PhaseCode algorithm, although we show through simulations that the Multicolor PhaseCode algorithm has slightly better performance in terms of smaller constants in the number of measurements needed. We now provide an illustrative toy example of the merge-and-color primitive operations in our Unicolor algorithm, which are sort of the conceptual dual to the peeling operations underlying the phase-aware systems. For now, we assume a genie is able to perform the following operations for us:

- A ball in a singleton bin can be detected with regard to its location and magnitude. Thus, this ball can be colored in our coloring algorithm. For example, if one has a measurement bin involving (x_1, x_2, x_3) and $x_2 = x_3 = 0$, the genie can tell us that this is a singleton measurement, the non-zero component is x_1, and recover its magnitude.

- Suppose that in a multiton bin, all the balls are colored, and the number of colors that are used is exactly two. In this case, each color corresponds to a local coordinate such that the the balls (non-zero components) with the same color are known relative to each other in phase and magnitude. Then, these two colors can be combined, that is all the non-zero components of that bin can be found relative to each other. For example, consider a measurement equation involving (x_1, x_2, x_3) where $x_3 = 0$ and x_1 and x_2 are also in singleton bins, then the genie can find x_1 and x_2 relative to each other, as well as their location indices.

- Suppose that in a multiton bin, only one ball is uncolored. Then, the genie can find the non-zero component corresponding to the uncolored ball with regard to its location, magnitude, and phase relative to the other colored balls. Thus, the uncolored ball gets colored. For example, consider a measurement equation involving (x_1, x_2, x_3), in which $x_2 = 0$, x_1 is known, and x_3 is unknown. Then, the genie can find the value and location of the non-zero component x_3.

Example Consider the left graph shown in Figure 3 that corresponds to a specific 4-sparse signal of length 6. The solid black balls represent non-zero components, and the blue dashed circles represent zero components of the signal. In the first iteration of our algorithm (See Figure 3), the singleton bins are found with the aid of a genie as discussed. In this example, the second bin containing x_3 and x_6 is a singleton since $x_6 = 0$. Thus, x_3 is colored, and it is recovered in magnitude. Without loss of generality, the phase of x_3 can be set to 0; thus, x_3 is fully uncovered. However, the edges connected to x_3 cannot be peeled from the other bins in our algorithm. Instead, with the aid of the genie, in the second iteration of the algorithm, x_1 is recovered in magnitude and phase relative to x_3, since the last bin contains only balls x_1 and x_3. Thus, x_1 also gets colored. In the third iteration, x_4 gets colored, since the first bin contains only x_1, x_4 and $x_5 = 0$. Thus, x_4 can be recovered in phase and magnitude. Finally, in the fourth iteration, x_2 gets colored, since the third bin contains x_3, x_4 (which are already colored), x_5, x_6 (which are zero), and x_2. Figure 3 illustrates the progress of our coloring algorithm in this example.

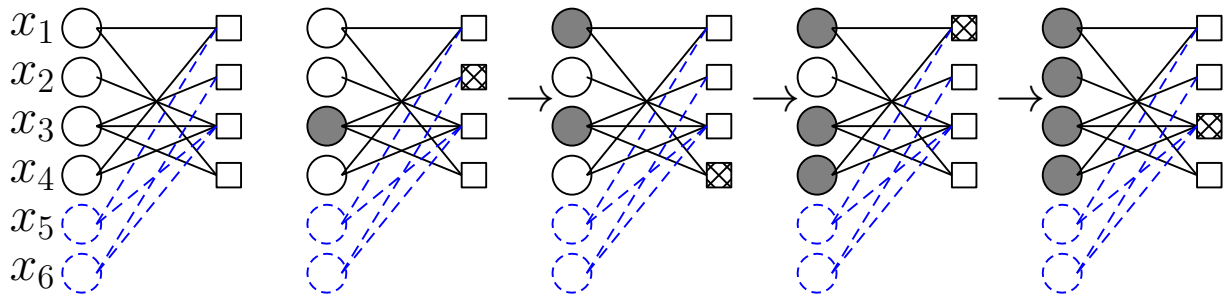


Figure 3: This figure shows a toy example illustrating the basics of Unicolor PhaseCode algorithm. At the first iteration of the algorithm, ball x_3 connected to the second bin which is a singleton, gets colored. In the following iterations, bins that contain only one uncolored ball corresponding to a non-zero component and some colored balls are detected, and that uncolored ball gets colored.

2.3 The PhaseCode Algorithm

Suppose that $x \in \mathbb{C}^n$ is exactly K-sparse. First we define $A \in \mathbb{C}^{4M \times n}$ to be a "row tensor product"[6] of matrices G and H, where $H \in \{0,1\}^{M \times n}$ is a binary "code" matrix and $G \in 4 \times n$ is the "trigonometric modulation" matrix that provides 4 measurements per each row of H. We define a row tensor product of matrices G and H, $G \otimes H$, as follows. Let $A = G \otimes H = [A_1^T, A_2^T, \ldots, A_M^T]^T$ and $A_i \in \mathbb{C}^{4 \times n}$. Then, $A_i(jk) = G_{jk} H_{ik},\ 1 \le j \le 4,\ 1 \le k \le n$.

Example Consider matrices

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

Then, our measurement matrix A is design from:

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

To illustrate the main idea of how to design the modulations, we provide a simple example here without going through details. Note that this is not the actual trigonometric-modulation matrix G used in PhaseCode, and is used for illustration purposes only. The goal is to shed light on how we can get rid of the genie assumed in Section 2.3. The details of the design of G will be described in Sections 2.5.

Example Suppose $x = [1, -2\mathbf{i}, 0, 0, 0]^T$ is a 2-sparse length-5 vector. Let $\omega = \frac{\pi}{10}$. Suppose the measure-

[6]Here, we do not follow popular convention for the notation for tensor product of matrices; instead, we define our own notation that is convenient for our purpose, and which should hopefully not cause any confusion.

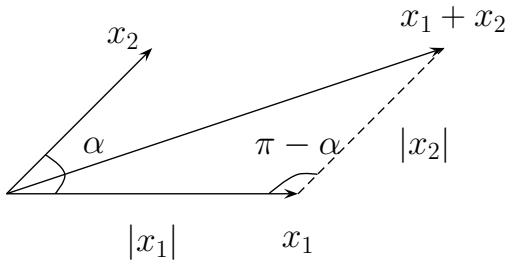


Figure 4: Recovering the angle between 2 vectors by the cosine law. The figure illustrates the cosine law: $|x_1+x_2|^2 = |x_1|^2 + |x_2|^2 + 2\cos(\alpha)$. One can compute $\cos(\alpha)$ using $|x_1 + x_2|$, $|x_1|$ and $|x_2|$. Thus, the only ambiguity about α is in the sign. This sign ambiguity can be resolved with an additional measurement as explained in the text.

ment matrix is

$$A = G \otimes H = \begin{bmatrix} 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & e^{\mathbf{i}\omega} & e^{\mathbf{i}2\omega} & e^{\mathbf{i}3\omega} & e^{\mathbf{i}4\omega} \\ 1 & \cos(\omega) & \cos(2\omega) & \cos(3\omega) & \cos(4\omega) \end{bmatrix}$$

$$= \begin{bmatrix} 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & e^{\mathbf{i}2\omega} & 0 & 0 \\ 1 & 0 & \cos(2\omega) & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & e^{\mathbf{i}\omega} & 0 & e^{\mathbf{i}3\omega} & 0 \\ 0 & \cos(\omega) & 0 & \cos(3\omega) & 0 \\ 1 & 1 & 0 & 0 & 1 \\ 1 & e^{\mathbf{i}\omega} & 0 & 0 & e^{\mathbf{i}4\omega} \\ 1 & \cos(\omega) & 0 & 0 & \cos(4\omega) \end{bmatrix}$$

In this example, we have 9 measurements in total comprising 3 sets of 3 each. Each set of 3 measurements corresponds to a single row of H. The measurements are as follows.

$$y_{11} = |x_1|,\ y_{12} = |x_1|,\ y_{13} = |x_1|,$$

$$y_{21} = |x_2|,\ y_{22} = |x_2 e^{\mathbf{i}\omega}|,\ y_{23} = |x_2\cos(\omega)|,$$

$$y_{31} = |x_1 + x_2|,\ y_{32} = |x_1 + x_2 e^{\mathbf{i}\omega}|,\ y_{33} = |x_1 + x_2\cos(\omega)|.$$

Now we describe how to mimic the genie explained in Section 2.3 using appropriate ratio tests and trigonometry based on these extra measurements. From the first set of 3 measurements, one can conclude that it is a singleton with high probability by observing $y_{11} = y_{12}$. But how can the decoder determine the location index of the singleton ball, as well as that of the other non-zero components? The ratio test $\frac{y_{13}}{y_{11}} = 1 = \cos(0)$ reveals that the non-zero component corresponding to the first set of measurements belongs to column 1. Note that if the non-zero component had belonged to column 2, the ratio test would have given $\cos(\omega)$ as the output. When $|x_1|$ is recovered, without loss of generality, its phase can be set to 0. Similarly, from the second set of measurements, singleton $|x_2|$ can be recovered by the second set of measurements.

It remains to find the relative angle between x_1 and x_2. Note that the decoder knows the measurement matrix, and so far it knows that x_1 and x_2 are non-zero components of x. Further, having access to the last set of 3 measurements which consist of x_1, x_2, and x_5, it can mimic the genie described earlier to resolve

Notation	Description
x	complex signal of length n
K	sparsity of signal
n	length of the signal
m	number of measurements
M	number of the rows of the code matrix
A	measurement matrix
H	code matrix
G	modulation matrix

Table 2: Table of Notations for Section 2.

the angle between x_1 and x_2. It does so by using a guess-and-check strategy of guessing that the last set of measurements involve only two non-zero balls, namely x_1 and x_2, and using y_{31} to find the relative angle of x_1 and x_2, α, up to a plus-minus sign uncertainty using the cosine law:

$$y_{31}^2 = y_{11}^2 + y_{21}^2 + 2y_{11}y_{21}\cos(\alpha).$$

See Figure 4 as an illustration. The correctness of the guess can be checked using the extra measurement y_{32} which can also be used to resolve the sign ambiguity of α if the guess is correct.

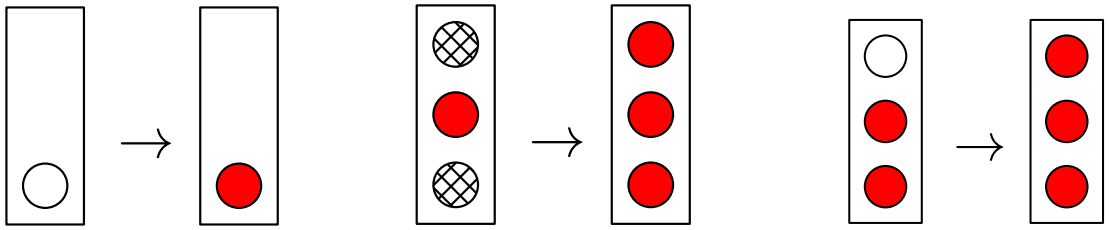
While we provide the details of how to design matrix G in Section 2.5, for completeness of the algorithm description, we state the precise expression for G without further explanation. Let $\omega' = \frac{2\pi L}{n}$ be a random phase between 0 and 2π, i.e. the discrete random variable L is uniformly distributed between 0 and $n-1$. We design $G \in \mathbb{C}^{4\times n}$ to be

$$G = \begin{pmatrix} e^{\mathbf{i}\omega} & e^{\mathbf{i}2\omega} & \dots & e^{\mathbf{i}n\omega} \\ e^{-\mathbf{i}\omega} & e^{-\mathbf{i}2\omega} & \dots & e^{-\mathbf{i}n\omega} \\ \cos(\omega) & \cos(2\omega) & \dots & \cos(n\omega) \\ e^{-\mathbf{i}\omega'} & e^{-\mathbf{i}2\omega'} & \dots & e^{-\mathbf{i}n\omega'} \end{pmatrix}. \tag{8}$$

Matrix H is constructed using a carefully chosen "balls-and-bin" model, where as mentioned, the balls refer to the non-zero values of x, and the bins refer to the measurements. Thus, each column of H denotes a ball and each row of H denote a bin. Matrix H is designed to ensure that each ball goes to exactly d bins uniformly at random. Thus, $H_{ij} = 1$ if and only if ball j is in bin i, and $H_{ij} = 0$ otherwise. Formally, we construct the ensemble of d-left regular degree bipartite graphs $\mathcal{C}^n(d, M)$, using a balls-and-bins model as follows. We construct a bipartite graph of n left nodes and M right nodes. When a ball goes to a bin, we construct an undirected edge between the corresponding left and right nodes in the bipartite graph.

Let $\tilde{x} \in \mathbb{C}^K$ be the vector that is constructed from the K non-zero components of x in the way that the order of these components' indices are maintained. Let $\tilde{H} \in \mathbb{C}^{M\times K}$ be the corresponding code matrix that is constructed from the active columns of H in the trivial way. Let $\mathcal{C}_1^K(d, m)$ be the ensemble of bipartite graphs induced by $\tilde{x}$. Note that the induced graph has also a d-left regular degree, and when K is large and M/K is a constant, the weight of each row of matrix $\tilde{B}$ or the right-node degree approaches a Poisson random variable with parameter $\lambda = \frac{Kd}{M}$.

We propose two decoding algorithms called Multicolor PhaseCode and Unicolor PhaseCode. We first describe Unicolor PhaseCode, and analyze it in Section 2.4. Next, we describe Multicolor PhaseCode that is more efficient algorithm, but whose analysis of remains unresolved, and is part of an ongoing work. Our decoding algorithms are based on the coloring of balls; and merging of different colors, that as mentioned earlier are fundamentally different from the "peeling-based" primitives underlying the decoding algorithms



(a) Coloring a ball in a singleton bin

(b) Combining colors in a multiton bin

(c) Coloring a ball in a multiton bin

Figure 5: This figure is showing the basic 3 merge-and-color primitives of our algorithm. Figure (a) illustrates that a ball in a singleton bin gets colored. Figure (b) illustrates that if a bin contains only colored balls of exactly two distinct colors, those colors can be combined. Figure (c) illustrates when a bin contains exactly one uncolored ball, and the other balls in the bin have all the same color, then the uncolored ball is colored with that color.

for phase-aware systems. With the aid of the carefully designed matrix G, our decoder is capable of performing the following functions:

- When a ball is in a singleton bin, that is a bin with only one ball, the ball is colored with a new color. Figure 5a illustrates this operation.
- When *all* the balls in a multiton bin (that is a bin with multiple balls) are colored, and the number of colors in that multiton bin is exactly two, then those two colors can be combined into a single composite color. Figure 5b illustrates this operation.
- When a multiton bin consists of exactly one *uncolored ball*, and the other non-empty set of balls in the bin have all the same color (let's say red), then the uncolored ball is colored with that color (i.e. it becomes red). Figure 5c illustrates this operation.

Unicolor Algorithm In the first iteration of the algorithm all the singletons are colored. In the second iteration, all the doubletons that each contain two colored balls from the first iteration, are detected, and their colors are combined. Then, the *largest* set of balls having the same color[7] is selected, and *every other ball gets uncolored*. At this point, there is only *one* color and no new colors are added to the system. Hence, we use the terminology *Unicolor* for this algorithm. In the following iterations, if there is only one uncolored ball in a bin, with one or more colored balls, then that uncolored ball gets colored. (See Figure 5c.) The algorithm continues until no more balls can be colored.

Multicolor Algorithm In the first iteration of the algorithm, all the singletons are colored. In the following iterations, the decoder checks all the non-singleton bins. If they consist of only 2 colors, those colors are combined. (See Figure 5b.) If the colored balls in the bin all have the same color, and if there is only one uncolored ball in that bin, then that ball gets colored. (See Figure 5c.) The algorithm continues until no more balls can be colored, or no more colors can be merged.

We provide pseudocode of both algorithms in Appendix C.

[7]Whenever two balls having colors C_1 and C_2 are combined, they get the same composite color C_{12}.

Remark Both algorithms have $\mathcal{O}(K)$ sample and decoding complexity, with Multicolor PhaseCode having a better constant than Unicolor PhaseCode in sample complexity due to its greater color-combining power.

Note that in both algorithms, the recovered balls are the *largest* set of balls having the same color. Intuitively, it is clear that the Unicolor PhaseCode is less expressive, since does not exploit the ability to combine colors after the second iteration. The following example illustrates the two algorithms and illustrates why Unicolor PhaseCode is suboptimal compared to Multicolor PhaseCode.

Example Let $K = 4$, $M = 5$ and $d = 2$. Label the balls by 1 to 4. Suppose that the induced bipartite graph is such that the bins are $\{1\}$, $\{1, 2\}$, $\{3\}$, $\{3, 4\}$, and $\{2, 3, 4\}$. In the first iteration, both algorithms color balls 1 and 3, let us say by red and blue, respectively since these balls are in singletons. In the second iteration, Multicolor PhaseCode can color ball 2 as red using $\{1, 2\}$, and ball 4 as blue using $\{3, 4\}$. However, the Unicolor PhaseCode does not find any doubleton containing balls 1 and 3. Thus, Unicolor PhaseCode has to pick either ball 1 and ball 3 randomly (let's say ball 1) as the largest set with one color. Finally, since bin 5 consists of a red ball and two blue balls, Multicolor PhaseCode has the ability to merge colors red and blue. This completes the successful decoding of Multicolor PhaseCode, since all the balls are colored and they have the same color. However, Unicolor PhaseCode can only color ball 2 and add it to the largest set. Thus, it recovers only 2 out of 4 balls.

The main theoretical contribution of this paper is the following theorem.

Theorem 2.1. *Let $A = G \otimes H$ be the measurement matrix, where H is chosen uniformly at random from the ensemble $\mathcal{C}^n(d, M)$ and G is the modulation matrix defined in* (8)*. Using the m measurements $y = |Ax|$, Unicolor PhaseCode is able to recover a fraction $1 - p^*(m)$ of nonzero components of x with probability $1 - \mathcal{O}(1/K)$, where m and $p^*(m)$ form a family of trade-offs as shown in Table 3 for selective operating points. In particular, Unicolor PhaseCode is able to recover a fraction $1 - 10^{-7}$ of nonzero components of x with $14K$ measurements with high probability. Furthermore, the decoding complexity of the algorithm is $\mathcal{O}(K)$ which is order-optimal.*

Proof. See Section 2.4. ■

The achievable trade-off between reliability and measurements cost as specified by Theorem 2.1 is shown in the following table.

m	$12.44K$	$12.72K$	$13.28K$	$\mathbf{13.92K}$	$14.64K$	$15.4K$
$p^*(m)$	1.1×10^{-3}	8×10^{-5}	3.2×10^{-6}	$\mathbf{1 \times 10^{-7}}$	2.9×10^{-9}	7×10^{-11}

Table 3: Family of trade-offs between error floor and number of measurements for Unicolor Phasecode. The table shows that to achieve higher reliability, i.e. smaller error floor, the number of measurements m should be increased.

Before we move to the proof of the main theorem, we first exhibit the simulated performance of Unicolor PhaseCode and Multicolor PhaseCode in Figure 6. Theorem 2.1 guarantees that Unicolor PhaseCode recovers a fraction $p^*(m)$ of x with m measurements with high probability, where $(m, p^*(m))$ can be chosen from Table 3. We chose the 3rd column of the table as an operating point, i.e., $(m, p^*(m)) = (13.28K, 3.2\times10^{-6})$. [8] Thus, we expect that the Unicolor PhaseCode algorithm will recover a fraction $1 - 3.2 \times 10^{-6}$ of K active symbols with high probability when $m = 13.28K$. Using the simulator, we measured error probability of both Unicolor PhaseCode and Multicolor PhaseCode while m is varied between $8K$ and $14K$; we ran each

[8] It will be explained in the following section how one can choose an operating point. For these simulations, we set the left degree as 7, i.e., $d = 7$.

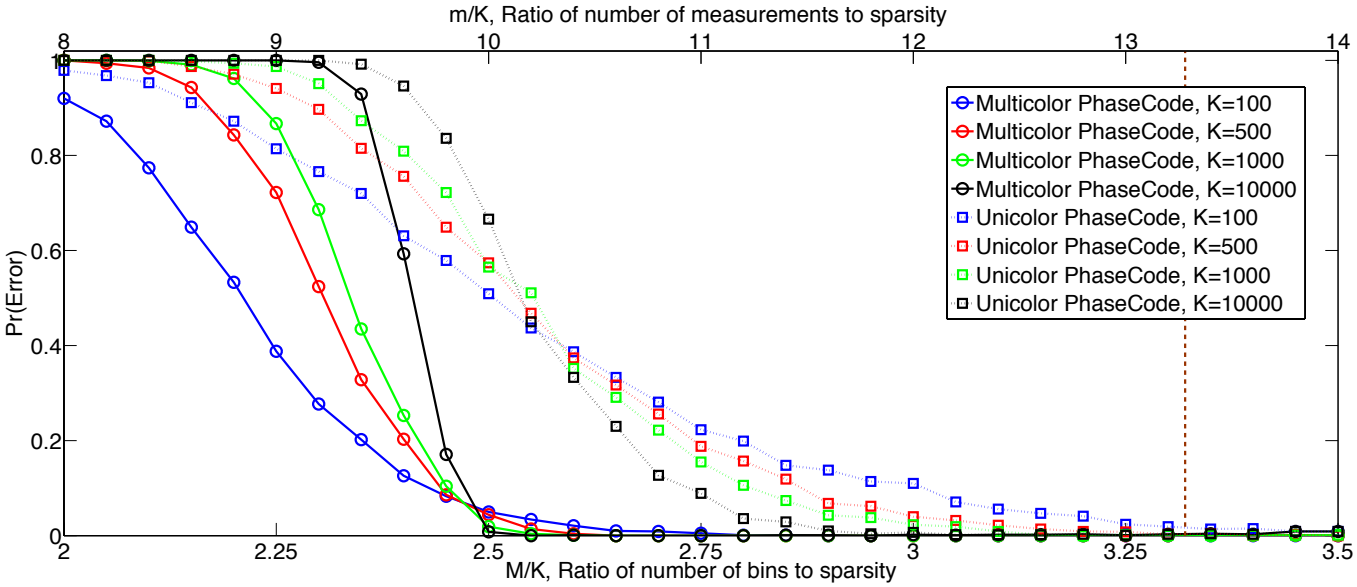


Figure 6: **Performance of PhaseCode Algorithms.** We evaluate Unicolor PhaseCode algorithm and Multicolor PhaseCode algorithm via simulations. We chose the 3rd column of the table as an operating point, i.e., $(m, p^*(m)) = (13.28K, 3.2 \times 10^{-6})$. Unicolor PhaseCode algorithm successsuccessfullyfully recovers almost all balls with very high probability when $m = 13.28K$. Multicolor PhaseCode is observed to achieve the same level or error probability with $m \simeq 11K$, that is about 17% reduction in number of measurements.

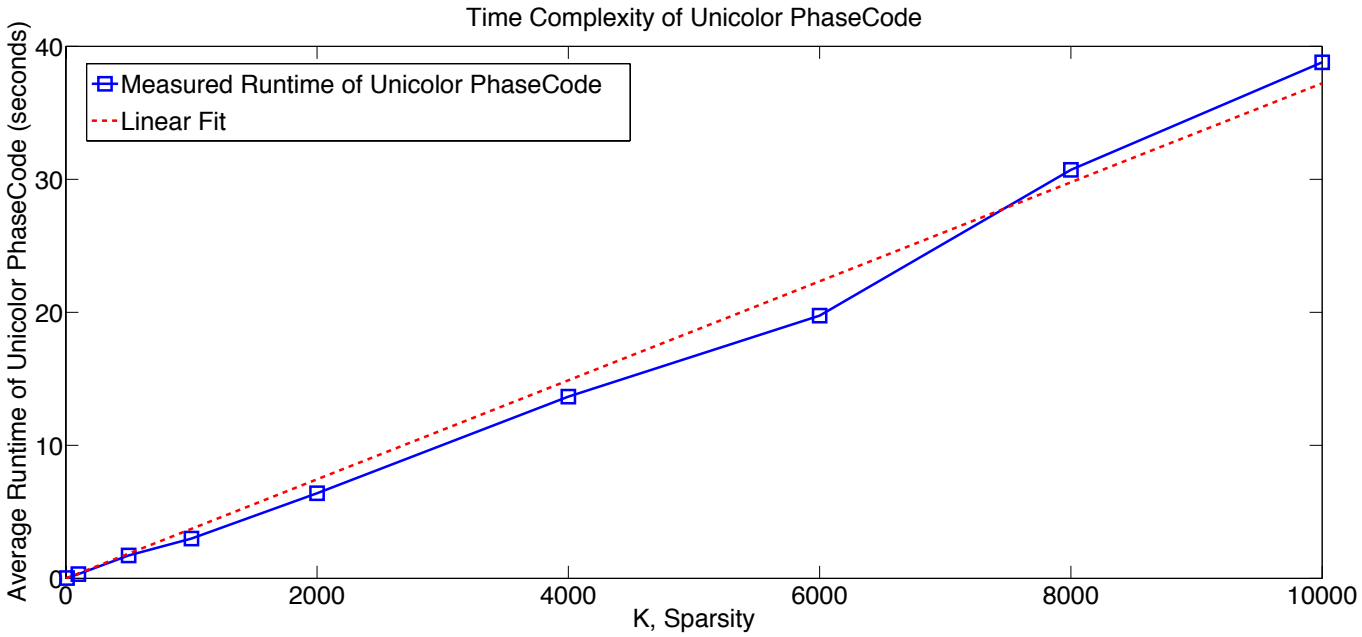


Figure 7: **Time Complexity of Unicolor PhaseCode.** We measured runtime of Unicolor PhaseCode algorithm. We chose $n = 10^{10}$ and varied K in order to see linear time complexity of the algorithm. Because both computational complexity and memory complexity depend only on K not n, we can easily simulate arbitrarily large n such as 10^{10}. Plotted is the average runtime of Unicolor PhaseCode algorithm, and it can observed that the average runtime of the algorithm increases linearly in K.

point 1000 times and determined error probability. Note that the error probability is defined as probability of not recovering a fraction $p^*(m)$ or more of nonzero components of x. We repeated the same set of simulations for several values for K.

As we claimed in the theorem, Unicolor PhaseCode algorithm succesfully recovers almost all balls with very high probability when $m = 13.28K$. It is also observed that the error probability of larger K is lower than that of smaller K. The simulation results not only support the main theorem but also show the superior performance of Multicolor PhaseCode algorithm over that of Unicolor PhaseCode algorithm: Multicolor PhaseCode is observed to achieve the same level or error probability with $m \simeq 11K$, that is about 17% reduction in number of measurements.

Theorem 2.1 states also that the decoding complexity of PhaseCode algorithms are $\mathcal{O}(K)$, which is order-optimal. In addition to that, its memory complexity is $\mathcal{O}(K)$, which is also order-optimal. In order to corroborate the claims, we measured the running time of Unicolor PhaseCode Algorithm. We chose the same operating point as in the above simulations: $(m, p^*(m)) = (13.28K, 3.2 \times 10^{-6})$. Indeed, we chose to add some measurements, $M = 14K$, in order to ensure a zero error probability. We randomly generated signals of length $n = 10^{10}$ using the Unicolor Algorithm. We increased the sparsity K up to 10^4 to see how the average runtime scales. The results are plotted in Figure 7; as K increases, the measured decoding time linearly increases; Unicolor PhaseCode successfully decodes $K = 10^4$ nonzero symbols from a signal of any length within 40 seconds. The exact runtime can be much improved considering that the simulator is written in Python and not fully optimized, and that we measured the runtime on a normal laptop.[9].

2.4 Analysis of Unicolor PhaseCode

In this section, we analyze the performance of Unicolor PhaseCode using *density evolution* techniques that are integral parts of modern coding theory. [5, 35]. Density evolution is a technique to analyze the performance of message passing algorithms on sparse-graph-codes. Density evolution computes the average message error probability of edges on the graph at each iteration of the algorithm. Our arguments are similar to that in [5]. 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 density evolution equation is however, different from the density evolution equation for phase-aware systems having similar graph ensemble, and requires a different analysis. Furthermore, our graph ensemble $\mathcal{C}_1^K(d, m)$ is different from the one in [5] as left nodes have regular degree, while right nodes have irregular degree in our ensemble. (In [5], both left and right nodes have regular degree.)

We now provide a brief outline of the proof elements (similar to the one provided in [6]), highlighting the main technical components needed to show that Unicolor PhaseCode recovers a fraction $1 - p^*(m)$ of the non-zero signal components with high probability.

- *Density evolution:* We analyze the performance of the Unicolor PhaseCode, over a typical graph of the ensemble $\mathcal{C}_1^K(d, m)$, for a fixed number of iterations, ℓ. First, we assume that a local neighborhood of depth 2ℓ of every edge in the graph is tree-like, i.e., cycle-free. Under this assumption, all the messages between balls and bins, in the first j iterations of the algorithm, are independent. Using this independence assumption, we derive a recursive equation that represents the expected evolution of the number of *uncolored* balls at each iteration.

- *Convergence to the cycle-free case:* : Using a Doob martingale as in [5], we show that the 2ℓ neighborhood of most of the edges of a randomly chosen graph from the ensemble is cycle-free with high probability. This proves that Unicolor PhaseCode decodes all but a small fraction of the left nodes with high probability in a constant number of iterations. The main difference of our convergence anal-

[9]For the measurements, we used a laptop with 2GHz Intel Core i7 and 8GB memory.

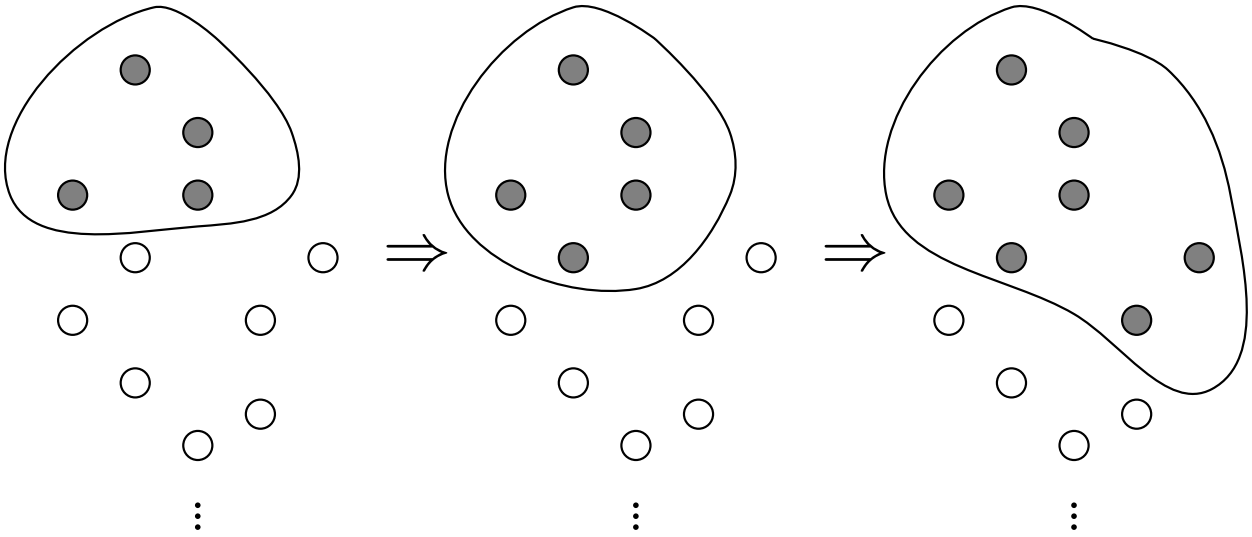


Figure 8: This figure illustrates that the giant component grows at each iteration of the Unicolor PhaseCode algorithm. The giant component keeps growing until almost all the balls are colored, that is almost all the balls will become part of the giant component.

ysis compared to [5] is that the right edge degree distribution in our ensemble is Poisson distributed, while the right degree is regular in [5].

Let $\rho_i,\ i \geq 1$ be the probability that a randomly selected edge has degree i on the right node. Since ρ_i is the fraction of edges that are connected to a right node of degree i, we have

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

Define $\rho(t) = \sum_{i=1}^{\infty} \rho_i t^{i-1}$ as the polynomial representing the edge degree distribution of right nodes. Then,

$$\rho(t) = \sum_{i\geq 1} \frac{\lambda^{i-1}e^{-\lambda}}{(i-1)!}t^{i-1} = \sum_{i\geq 0} \frac{\lambda^i e^{-\lambda}}{i!}t^i = e^{-\lambda}e^{\lambda t} = e^{-\lambda(1-t)}. \tag{9}$$

Let $\mu_i,\ i \geq 1$ be the probability that a randomly selected edge has degree i on the left node. Clearly, $\mu_i = 1_{\{i=d\}}$, where 1_E is the indicator that event E has happened, i.e. $1_E = 1$ if E is true, and $1_E = 0$, otherwise. Define $\mu(t) = \sum_{i=1}^{\infty} \mu_i t^{i-1} = t^{d-1}$ be the polynomial representing the edge degree distribution of left nodes.

At each iteration of Unicolor PhaseCode, we call the giant component as the largest set of balls that have the same color. The algorithm follows 3 major steps to recover almost all the balls by coloring them.

- *Step* 1*:* All the singleton bins and their corresponding balls are found.
- *Step* 2*:* The giant component is formed by finding doubleton bins having both balls in a singleton.
- *Step* 3*:* After the giant component is formed, at each iteration of the algorithm, more balls are colored and connected to the giant component. See Figure 8 for an illustration of this step.

Let p_j be the probability that a randomly chosen ball does not belong to the giant component at step j. The density evolution equation is an equation relating p_j to p_{j+1}. Under the tree-like assumption, and for $j \geq 2$ one has

$$p_{j+1} = (1 + e^{-\lambda} - e^{-\lambda p_j})^{d-1}. \tag{10}$$

Here is a proof of Equation (10). A ball v passes a message to bin c that it is not part of a giant component at step $j+1$, if none of the other $d-1$ neighbor bins of v can tell v that it is part of the giant component at step j. First note that if a bin is a singleton, it cannot tell the ball that it is part of a giant component. This is a fundamental difference of our decoding process compared to that of conventional peeling-based decoders such as the LDPC decoder for erasure channel [35]. In LDPC decoding, since there is no phase ambiguity, as soon as a singleton bin is detected, the ball in the singleton bin is recovered and it is peeled from all other bins that also contain that ball. However, singleton balls cannot be peeled from other bins in our setting. Indeed, our problem has the peculiar attribute that singleton bins, while critical to initiating the growth of the giant component at the outset, are not useful once a giant component is formed, and too many singletons actually hurt the system performance by featuring useless isolated measurements. This is a significant departure from "phase-aware" measurement system like LDPC codes.

More precisely, a bin can tell a ball that it is not part of the giant component if the bin is connected to a non-empty set of balls other than v, and they are all in the giant component. This happens with probability

$$\begin{aligned}\sum_{i=2}^{\infty} \rho_i (1-p_j)^{i-1} &= \rho(1-p_j) - \rho_1 \\ &= e^{-\lambda p_j} - e^{-\lambda}.\end{aligned}$$

Thus, the probability that ball v passes a message to bin c that it is not part of the giant component is the

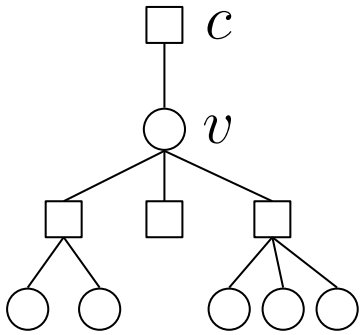


Figure 9: Length-2 tree-like neighborhood of (v, c) for $d = 4$. The neighborhood is the subgraph of all the edges and nodes of paths having length less than or equal to 2, that start from v and the first edge of the path is not (v, c).

probability that none of the other $d-1$ bins can tell v that it is in the giant component. See Figure 9 for an illustration of the case $d = 4$. These messages are all independent if the bipartite graph is a tree. Assuming this, one has

$$p_{j+1} = (1 - (e^{-\lambda p_j} - e^{-\lambda}))^{d-1}.$$

Note that in Multicolor PhaseCode, another possibility of joining the giant component is that v is colored, let's say as red, with the color of the giant component being another color, say blue, and a neighbor bin of v contains only blue and red balls. This will boost system performance by accelerating the coloring and merging process, but is hard to analyze precisely. Therefore, for analytical purposes only, we do not allow this opportunity to be exploited by Unicolor PhaseCode.

An interesting but unfortunate fact is that $p_0 = 1$ is a fixed point of the density evolution equation. Thus, one cannot use (10) at the outset to follow the evolution of p_j, and to argue that it goes close to 0, since p_j can get stuck at 1. To use Equation (10), we need a more careful characterization of the first two steps of the algorithm that form the giant component. At the first iteration, all the balls in singleton bins are found. Therefore, no giant component is yet formed; thus, $p_1 = 1$. At the second iteration, the giant component is formed by coloring the balls in doubleton bins having both balls in singleton bins found in step 1. After the giant component is formed in the second iteration, the probability that a randomly chosen ball is part of the giant component is p_2. If one can show that p_2 is small enough such that after a fixed number of iterations p_j gets close to 0, then concentration bounds can be used to show that the number of balls not being in the giant component is indeed highly concentrated around its mean after ℓ iterations, Kp_ℓ. In Lemma 2.4, we show that if $p_2 = 1 - \delta$ for some constant $0 < \delta < 1$ independent of K, p_j gets close to 0 after a constant number of iterations. Clearly $p_2 = 1 - \delta$ if there exists a giant component of size linear in K after the second step.

Towards this end, in Lemma 2.2, we form a graph with nodes that are balls which are in singletons. We consider edges between these balls if they are in a doubleton, and we use an Erdos-Renyi random graph model [36] to find parameters d and M for which there is a giant component of size linear in K in the initial phase of the algorithm. The Erdos-Renyi random graph model is characterized by 2 parameters: n which is the number of nodes in the graph and p which is the probability that each of the $\binom{n}{2}$ possible edges are connected. Note that each edge is included in the graph with probability p independent from every other edge. There is another variant of Erdos-Renyi random graph model which is parametrized by (n, M) where M is the total number of edges. Then, the graph is chosen uniformly at random from the collection of all graphs with n nodes and M edges. By the law of large numbers, the two models are equivalent for $M = \binom{n}{2}p$ as long as $n^2 p \to \infty$. It is well known that in an ER model if $np \to c > 1$, as $n \to \infty$, where c is some constant, then the graph will have a unique giant component of size linear in n [36].

Define K_s to be the random variable representing the number of balls that are in singletons. We form an Erdos-Renyi random graph model with parameters (K_s, p_s) or equivalently parameters (K_s, M_s) where p_s is the probability that an edge is connected, and M_s is the total number of edges. Thus, as K_s gets large, M_s becomes $\binom{K_s}{2}p_s$. Now we compute the parameters K_s and p_s as follows. The probability of a ball being in a singleton is the probability that at least one of its d neighbor bins is a singleton bin, that is:

$$q_s = 1 - (1 - \rho_1)^d. \tag{11}$$

Thus, by the law of large numbers as K gets large, there are about Kq_s distinct balls in singleton bins. Let $M = cK$ for some constant c. As K gets large, the number of doubleton bins becomes $M\frac{\lambda^2 e^{-\lambda}}{2!}$ since the number of balls in a bin is a Poisson random variable with parameter λ. However, we are interested only in distinct doubletons. It is easy to see that as K gets large, essentially all but a vanishing fraction of the doubleton bins are distinct. To this end, fix a doubleton (v_1, v_2). The probability that a randomly chosen doubleton bin contains (v_1, v_2) is $1/\binom{K}{2}$. The number of doubleton bins is linear in K; thus only a vanishing $\mathcal{O}(1/K)$ fraction of them are non-distinct.

We now do a careful analysis. Let M_s be the number of doubleton bins for which both balls are also in other singleton bins. Thus, M_s is the number of edges of the Erdos-Renyi graph by construction. Consider a random ball i. Let D be the event that i is in a doubleton bin and S be the event that i is in a singleton bin.

We compute the following 2 relevant conditional probabilities:

$$\begin{aligned} p_1 \triangleq \mathbb{P}(D|S) &= \frac{\mathbb{P}(D \cap S)}{\mathbb{P}(S)} \\ &= \frac{1 - \mathbb{P}(\bar{S}) - \mathbb{P}(\bar{D}) + \mathbb{P}(\bar{S} \cap \bar{D})}{1 - \mathbb{P}(\bar{S})} \\ &= \frac{1 - (1-\rho_1)^d - (1-\rho_2)^d + (1-\rho_1-\rho_2)^d}{1-(1-\rho_1)^d}. \\ p_2 \triangleq \mathbb{P}(D|\bar{S}) &= 1 - \mathbb{P}(\bar{D}|\bar{S}) \\ &= 1 - \frac{\mathbb{P}(\bar{S} \cap \bar{D})}{\mathbb{P}(\bar{S})} \\ &= 1 - \frac{(1-\rho_1-\rho_2)^d}{(1-\rho_1)^d}. \end{aligned}$$

Now we use Bayes' rule to find that

$$q \triangleq \mathbb{P}(S|D) = \frac{\mathbb{P}(D|S)\mathbb{P}(S)}{\mathbb{P}(D|S)\mathbb{P}(S) + \mathbb{P}(D|\bar{S})\mathbb{P}(\bar{S})} = \frac{p_1 q_s}{p_1 q_s + p_2(1-q_s)}.$$

Thus,

$$M_s = M \frac{\lambda^2 e^{-\lambda}}{2!} q^2. \tag{12}$$

We construct an Erdos-Renyi graph with $K_s = K(1-(1-\rho_1)^d)$ nodes and M_s edges chosen uniformly at random among $\binom{K_s}{2}$ possible edges. The probability of a randomly chosen edge being connected is thus:

$$p_s = \frac{M \frac{\lambda^2 e^{-\lambda}}{2!} q^2}{\binom{K_s}{2}}.$$

In the following lemma, we address, given a particular choice of parameter d, for what values of M, a giant component of size linear in K (or equivalently K_s) is formed. We pick 2 particular left degrees: $d = 5$ and $d = 8$. As we will see, choosing $d = 5$ is optimal in terms of having the smallest number of required measurements. [10] However, picking $d = 5$ is not optimal in terms of having the smallest possible error floor. Indeed, as d increases the error floor decreases. (See Table 4.) Therefore, depending on the required reliability, one can first pick parameter d, and then pick the required number of measurement for that parameter. Choosing $d = 8$ leads to recovering almost all the non-zero components but a small fraction of 10^{-7} of them with only $14K$ measurements. Note that these are only particular choices from a whole family of trade-offs. We picked $d = 8$ for highlighting our result as it represents a good trade-off between reliability and measurements cost.

Lemma 2.2. *If $d = 5$ and $3.11K \leq M \leq 19.24K$, with probability $1 - \mathcal{O}(1/K)$, there exists a giant component of size linear in K formed by the balls in singletons. For $d = 8$, there is a giant component with high probability if $3.48K \leq M \leq 55.36K$.*

Proof. From the well-known Erdos-Renyi random graph result [36] (also see [1]), a linear size giant component exists if $K_s p_s > 1$ with probability $1 - \mathcal{O}(1/K_s)$. More precisely, let Z be the size of the giant component. Then, one has

$$\mathbb{P}\left(|\frac{Z}{K_s} - \zeta| < \varepsilon\right) = 1 - O\left(\frac{1}{\varepsilon^2 K_s}\right),$$

[10] It is interesting to contrast this with the phase-aware left-regular sparse-graph-codes case, where the optimal choice is $d = 3$ [6].

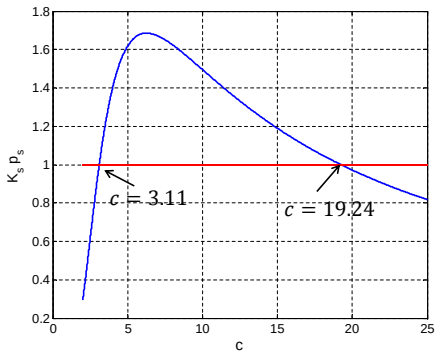


Figure 10: The diagram is showing for what values of c the giant component is formed after step 2 of the algorithm. Note that $c = M/K$. In the random graph model the giant component is form if $K_s p_s > 1$, where K_s is the number of nodes in the random graph, and p_s is the probability that an edge is connected. From the diagram, one can see that if $3.11 < c < 19.24$, the condition for having a giant component is satisfied.

where $\zeta \in (0, 1)$ is the unique solution of $\zeta + e^{-2\zeta M_s/K_s} = 1$, if $2M_s/K_s > 1$ or equivalently $K_s p_s > 1$ [1,2]. Thus, a linear-size giant component exists if

$$\frac{K q_s M_s}{\binom{K q_s}{2}} > 1.$$

Let $d = 5$. Replacing M_s and q_s by (12) and (11), one can check that the inequality holds if $3.11 \leq c \leq 19.24$ (See Figure 10). Similarly, one can set $d = 8$ and see that the inequality holds if $3.48 \leq c \leq 55.36$. ■

Recall that p_j is the probability that a randomly chosen ball is not part of the giant component at iteration j. By Lemma 2.2, for proper choices of d and M, there exists a linear-size giant component, let's say of size $K_s = \delta K$ for some constant $0 < \delta < 1$, at step 2. Therefore, $p_2 = \frac{K - K_s}{K} = 1 - \delta$. The following corollary is an immediate result of Lemma 2.2.

Corollary 2.3. *There exists a constant* $0 < \delta < 1$ *independent of* K*, such that* $p_2 = 1 - \delta$.

Due to the formation of a linear-size giant component in step 2 of the algorithm, we can revisit the density evolution equation (10):

$$p_{j+1} = (1 + e^{-\lambda} - e^{-\lambda p_j})^{d-1},$$

with the aid of Corollary 2.3, which guarantees that p_2 is strictly smaller than 1. Recall that $p_0 = 1$ is a fixed point of (10). But with the giant component formation, we can break away from the shackles of "being stuck" at $p_0 = 1$. With $p_2 < 1$, we hope to find a better fixed point of (10) to which our density evolution will converge.

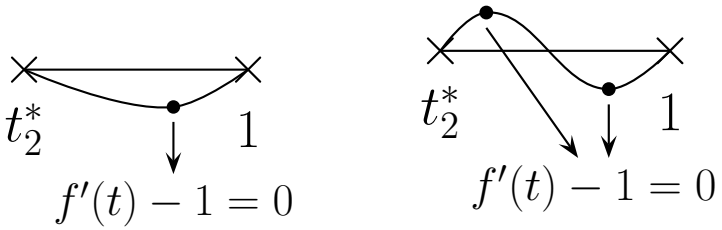
Towards this end, ideally one wants Equation (2.3) to have the property

$$p_{j+1} = (1 + e^{-\lambda} - e^{-\lambda p_j})^{d-1} < p_j, \tag{13}$$

for all $p_j \in (0, 1)$. Let's take a closer look at the fixed point equation

$$t = f(t) = (1 + e^{-\lambda} - e^{-\lambda t})^{d-1}. \tag{14}$$

As mentioned, one solution is $t_1^* = 1$. As we can break away from t_1^*, fortunately there exists another solution approximately at $t_2^* \simeq e^{-\lambda(d-1)}$ which is close to 0. To see this, consider the equation $y = (1 +$



(a) The good case. (b) The bad case.

Figure 11: Figure (a) illustrates the good case that there are no fixed points other than 1 and t_2^*. Figure (b) illustrates the bad case that there is another fixed point in the interval $(t_2^*, 1)$. In this case, $f'(t) = 1$ has two solution for $t \in (t_2^*, 1)$, as it is shown in Figure (b).

$e^{-\lambda} - e^{-\lambda x})^{d-1}$. Suppose that $0 < x = e^{-\lambda(d-1)} \ll 1$. Then, $e^{\lambda x} \simeq 1$ and $1 + e^{-\lambda} - e^{-\lambda x} \simeq e^{-\lambda}$. Thus, $y = x$ which shows that $e^{-\lambda(d-1)}$ is approximately another fixed point of (10). [11] From now on, we will refer to this fixed point as the error floor p^*.

Lemma 2.4. *Let $d = 5$. If $2.33K \leq M \leq 13.98K$, then the fixed point equation* (14) *has exactly 2 solutions for $t \in [0, 1]$: $t_1^* = 1$ and $t_2^* \simeq e^{-\lambda(d-1)}$ (See Figure 12). For $d = 8$, a similar result holds if $2.63K \leq M \leq 47.05K$.*

Proof. First, let's consider a small neighborhood around $t_1^* = 1$. We want

$$f(t_1^* - h) < t_1^* - h = f(t_1^*) - h,$$

for some small $h > 0$. Equivalently, we want

$$\frac{f(t_1^*) - f(t_1^* - h)}{h} > 1.$$

Letting $h \to 0$, the condition becomes $f'(t)|_{t=1} > 1$. This is a necessary and sufficient condition for instability of point $t = 1$. In other words, this condition makes sure that (13) holds for p_j close to 1. Thus, in picking parameters d and λ, one makes sure that

$$f'(t)|_{t=1} = (d-1)\lambda e^{-\lambda} > 1.$$

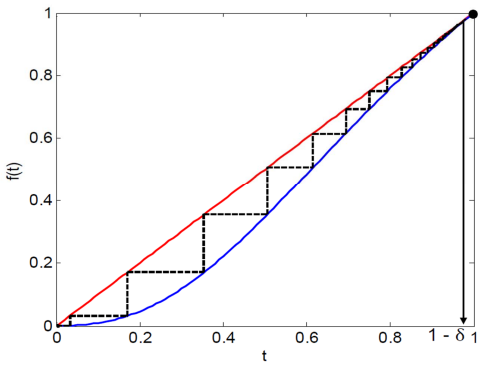
For $d = 5$, this leads to $0.3574 < \lambda < 2.1533$ or $2.32K < M < 13.99K$. For $d = 8$, this leads to $0.17 < \lambda < 3.06$ or $2.62K < M < 47.06K$. To complete the proof, we need to show that $f(t) - t < 0$ for $t \in (t_2^*, 1)$. Note that $f(t)$ is continuous and continuously differentiable. Thus to show that $f(t) - t < 0$ for $t \in (t_2^*, 1)$, it is enough to show that $f'(t) - 1 = 0$ has only one solution in that interval (the "good" case: See Figure 11a). To see this, suppose that $f(t) - t = 0$ for some t in the interval $(t_2^*, 1)$. Since 1 and t_2^* are also solutions of $f(t) - t = 0$, then $f'(t) - 1$ must change sign at least twice in the interval $(t_2^*, 1)$ (the "bad" case: See Figure 11b). Therefore, to ensure that $f(t) < t,\ \forall t \in (t_2^*, 1)$ it is sufficient to show that

$$f'(t) = \lambda e^{-\lambda t}(d-1)(1 + e^{-\lambda} - e^{-\lambda t})^{d-2} = 1,$$

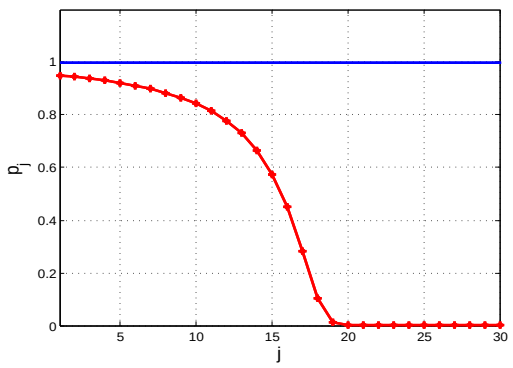
has only one solution in the interval $t \in (t_2^*, 1)$. After some algebra, one can re-write the above equation as

$$(\lambda(d-1))^{-\frac{1}{d-2}} e^{\lambda t(1/(d-2)+1)} = e^{\lambda t}(1 + e^{-\lambda}) - 1,$$

or an equation of the form $e^{az} = be^z - c$ for $a > 1$ and $b, c > 0$ which has clearly at most one solution since the exponent of one of the exponential terms is larger. ■



(a) The density evolution curve for parameters $d = 5$ and $\lambda = 2$.



(b) The evolution of p_j after each iteration for $d = 5$ and $\lambda = 2$.

Figure 12: 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 $p_0 = 1 - \delta$, p_j decreases after each iteration, and it gets very close to 0. Figure (b) illustrates the same phenomenon by showing the evolution of p_j versus the iteration, j. Note that in this example, p_j gets very close to 0 after only 20 iterations.

With the aid of Lemma 2.4, we can show that p_j gets very close to fixed point $p^* = t_2^*$ after a constant number of iterations. This is established in the following corollary.

Corollary 2.5. *For any $\epsilon > 0$, there exists a constant $\ell(\epsilon)$ such that $p_\ell \leq p^* + \epsilon$.*

Proof. By Lemma 2.4, p_j decreases at each iteration of the algorithm. Thus, $p_j,\ j \geq 1$ is a decreasing sequence which is lower bounded by p^*. Thus, it will converge to p^*. However, convergence to p^* is not sufficient for us. To formally prove the corollary, we need to show that after each iteration, the probability of not being part of the giant component decreases by a constant amount, that is a function of ϵ but is not a function of K. Let $p_m(\epsilon) = \arg\min_{p \in [p^* + \epsilon, p_2]} p - f(p)$. Let $\eta(\epsilon) = p_m - f(p_m)$. Then,

$$p_{j+1} - p_j = f(p_j) - p_j \leq -\eta.$$

Therefore, it takes at most $\ell(\epsilon) = \frac{1-p^*}{\eta} < \frac{1}{\eta}$ iterations to reach $p^* + \epsilon$. ■

Table 4 illustrates how the error floor p^* and the minimum ratio of bins to balls $c = M/K$ change for different values of d. If our reliability target allows the error floor to be set at 1.1×10^{-3}, then $d = 5$ minimizes the number of required bins. Recall that the total number of measurements is $m = 4M$ which matches the result of Table 3. (See Section 2.3) If one wants to achieve smaller error floor, then d and c should be both increased.

d	4	**5**	6	7	8	9	10
p^*	2.7×10^{-2}	1.1×10^{-3}	8×10^{-5}	3.2×10^{-6}	1×10^{-7}	2.9×10^{-9}	7×10^{-11}
c	3.31	**3.11**	3.18	3.32	3.48	3.66	3.85

Table 4: The table shows how error floor, p^*, and $c = M/K$ (which indirectly determines the number of measurements) vary for different values of left degree, d. The minimum value of c is 3.31 that is achieved when $d = 5$. Moreover, one can see that p^* decreases as d increases.

[11] Of course, one can easily find the exact solution to (14), using numerical methods for given values of d and λ.

In the density evolution analysis so far, we have shown that the *average* fraction of balls that cannot be recovered 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. Towards this end, first in Lemma 2.6 we show that a neighborhood of depth ℓ of a typical edge is a tree with high probability for a constant ℓ. Second, in Lemma 2.7, we use the standard Doob's martingale argument [5], to show that the number of uncolored balls after ℓ iterations of the algorithm is highly concentrated around Kp_ℓ.

Consider a directed edge from $\vec{e} = (v, c)$ from a left-node (ball) v to a right-node (bin) c. Define the directed neighborhood of depth ℓ of $(\vec{e})$ as $\mathcal{N}_{\vec{e}}^{\ell}$, that is the subgraph of all the edges and nodes on paths having length less than or equal to ℓ, that start from v and the first edge of the path is not $\vec{e}$. As an example, the directed neighborhood of depth 2 of $(\vec{e})$ is shown in Figure 9.

Lemma 2.6. *For a fixed ℓ^*, $\mathcal{N}_{\vec{e}}^{2\ell^*}$ is a tree-like neighborhood with probability at least $1 - \mathcal{O}(\log(K)^{\ell^*}/K)$.*

The proof is provided in Appendix A.

Lemma 2.7. *Over the probability space of the ensemble of graphs $\mathcal{C}_1^K(d, m)$, let Z be the number of uncolored edges after ℓ iterations of the Unicolor PhaseCode algorithm. Then, for any $\epsilon > 0$, there exists a large enough K and constants β and γ such that*

$$|\mathbb{E}[Z] - Kdp_\ell| < Kd\epsilon/2 \tag{15}$$

$$\mathbb{P}(|Z - Kdp_\ell| > Kd\epsilon) < 2e^{-\beta\epsilon^2 K^{1/(4\ell+1)}}, \tag{16}$$

where p_ℓ is derived from the density evolution equation (10)*.*

The proof is provided in Appendix B.

Now gathering the results of Corollary 2.5 and Lemmas 2.2 and 2.7 completes the proof of Theorem 2.1. Note that the dominant probability of error is due to the event that the giant component is not formed in the second iteration which happens with probability $\mathcal{O}(1/K)$. It is worth mentioning that Lemma 2.6 is only used to prove Lemma 2.7. Thus, the event that an edge does not have a tree-like neighborhood, which happens with some probability upper bounded by $\mathcal{O}(\frac{\log(K)^{\ell^*}}{K})$, is not an error event of the algorithm.

2.5 Measurement Design: "Trig-Modulation"

In this section, we will explain the choice of the measurement matrix G. Our design of G draws heavily from the proposed trigonometric subsystem in [2] with proper modifications to better match our sparse-graph-code subsystem, H, that is distinct from [2]. We also show that one can decrease the number of these trig-based measurements from 5 per bin as proposed in [2] to 4 per bin as we describe.

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 $\omega = \frac{\pi}{2n}$. We design the measurement matrix $G = [g_{j\ell}]$ as follows. For all $\ell,\ 1 \leq \ell \leq n$,

$$\begin{aligned} g_{1\ell} &= e^{\mathbf{i}\omega\ell}, \\ g_{2\ell} &= e^{-\mathbf{i}\omega\ell}, \\ g_{3\ell} &= 2\cos(\omega\ell), \\ g_{4\ell} &= e^{\mathbf{i}\omega'\ell}, \end{aligned}$$

where as mentioned in Section 2.3, $\omega' = \frac{2\pi L}{n}$ and L is uniformly distributed between 0 and $n-1$.

As mentioned in Section 2.3, the measurement matrix should enable us to detect whether we have a singleton bin, as well as, if yes, the location index of the corresponding ball in the singleton bin. Furthermore,

it should detect if a multiton bin consists of only known balls having exactly two unique colors. We call these as mergeable multitons (See Figure 5b). Finally, if a multiton bin consists of known balls with the same color and only one uncolored ball, the measurement should be able to find the index of uncolored ball. We call these as resolvable multitons as in [2] (See Figure 5c). In the following, we show how each of these detections can be accomplished using "guess and check" approach. We provide pseudocode of these bin processors in Appendix C.

(i) Singletons: Suppose that we want to check the hypothesis that the i-th bin is a singleton. If the bin is a singleton, only one non-zero component of x, let's say x_ℓ, is involved in vector y_i, that is $y_{i,1} = |x_\ell e^{\mathbf{i}\omega\ell}|$, $y_{i,2} = |x_\ell e^{-\mathbf{i}\omega\ell}|$, and so on. Thus, the i-th bin is a singleton only if $y_{i,1} = y_{i,2} = y_{i,4}$. The event that bin i is not a singleton, and all these measurements are equal has measure 0 for our generic choice of signal components. In order to find the index ℓ, one uses $y_{i,3}$ to get

$$\ell = \frac{1}{\omega}\cos^{-1}(\cos(\omega\ell)) = \frac{1}{\omega}\cos^{-1}(\frac{y_{i,3}}{2y_{i,1}}).$$

Note that $\cos(\omega\ell)$ is positive if $0 \leq \omega \leq \frac{\pi}{2n}$ for all $\ell,\ 1 \leq \ell \leq n$.

(ii) Mergeable multitons: Consider a bin i as in Figure 5b, in which we already know that there are some red balls (non-empty set $\mathcal{R}$) and some blue balls (non-empty set $\mathcal{B}$). This means that the red balls are known in magnitude and phase relative to each other, and similarly the blue balls are known relative to each other. However, the relative phase of blue balls and red balls are not known. If there is no other ball in the bin, we show that the relative phase can be found. Thus, the colors can be combined (We again deploy a guess and check strategy.). First, we guess that bin i has no balls other than the two sets of colored ones. Then, we have access to measurement

$$y_{i,1} = |r + b|,$$

where $r = \sum_{\ell\in\mathcal{R}} x_j e^{\mathbf{i}\omega\ell}$ is the sum of complex numbers corresponding to the red balls, and $b = \sum_{\ell\in\mathcal{B}} x_\ell e^{\mathbf{i}\omega\ell}$ is the sum of complex numbers corresponding to the blue balls. Since red balls are known up to a local phase, $|r|$ is known. Similarly, $|b|$ is also known. Without loss of generality pick some $\ell_r \in \mathcal{R}$ and set the phase of x_{ℓ_r} to 0 to form the local coordinate for red balls. Furthermore, pick some $\ell_b \in \mathcal{B}$ and set the phase of x_{ℓ_b} to 0 to form the local coordinate for blue balls. Given the local, coordinates $r = |r|e^{\mathbf{i}\phi_r}$ and $b = |b|e^{\mathbf{i}\phi_b}$ are known. By the cosine line, the true relative phase between r and b can be found as

$$\theta = \cos^{-1}(\frac{|r|^2 + |b|^2 - y_{i,1}^2}{2|r||b|}), \tag{17}$$

up to a plus-minus sign. Assuming that the plus sign is true, we can merge these balls as follows. Without loss of generality, we set the phase of x_{ℓ_r} to 0. Thus, $r = |r|e^{\mathbf{i}\phi_r}$ and $b = |b|e^{\mathbf{i}(\phi_r+\theta)}$. This shows that the local coordinate in $\mathcal{B}$ should be rotated by an angle $\theta + \phi_r - \phi_b$ to match with the new global coordinate. Hence, we recover all the blue balls with respect to the coordinate of red balls, and the colors can be combined. A similar procedure can be done for the solution of θ with a minus sign. Now we again use the check equation to find whether one of these relative phases works. If none of them works, our guess is wrong, and bin i is not mergeable. Thus, we need to check whether

$$|\sum_{\ell\in\mathcal{R}\cup\mathcal{B}} x_\ell e^{\mathbf{i}\omega'\ell}| = y_{i,4}$$

is satisfied or not for the 2 values of θ derived in (17). If the guess is correct, the probability that the check fails is 0; thus, one can recover the resolvable multiton with probability 1.

(iii) Resolvable multitons: Consider a bin, let's say bin i, in which we know that there are some known balls that have the same color. We want to check if bin i has exactly one other uncolored ball; i.e. one unknown non-zero component of x, say x_ℓ, as in Figure 5c. We now describe our guess and check strategy to check if bin i is indeed a resolvable multiton, and if so, to find ℓ and x_ℓ. If our guess is correct, we have access to measurements of the form:

$$y_{i,1} = |a + e^{\mathbf{i}\omega\ell} x_\ell| = |u|, \tag{18}$$

$$y_{i,2} = |b + e^{-\mathbf{i}\omega\ell} x_\ell| = |v|, \tag{19}$$

$$y_{i,3} = |c + 2\cos(\omega\ell) x_\ell| = |w|, \tag{20}$$

$$y_{i,4} = |d + e^{\mathbf{i}\omega'\ell} x_\ell|, \tag{21}$$

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 (18)-(20) to find ℓ and x_ℓ, and use (21) to check if our guess is correct. Since $e^{\mathbf{i}\omega\ell} + e^{-\mathbf{i}\omega\ell} = 2\cos(\omega\ell)$, we know that $u + v = w$. Let α be the angle between complex numbers u and v. Then,

$$|u+v|^2 = |u|^2 + |v|^2 + 2|u||v|\cos(\alpha).$$

Thus, one can find α up to a plus-minus sign as,

$$\begin{aligned}\alpha &= \cos^{-1}\left(\frac{|u+v|^2 - |u|^2 - |v|^2}{2|u||v|}\right) \\ &= \cos^{-1}\left(\frac{y_{i,3}^2 - y_{i,1}^2 - y_{i,2}^2}{2y_{i,1}y_{i,2}}\right).\end{aligned}$$

We find possible x_ℓ's for two different signs of α. If our guess is true, the check measurement $y_{i,4}$ will determine which solution is the right one. Define a known variable z as

$$z = u/v = \frac{|u|}{|v|} e^{\mathbf{i}\omega\alpha}.$$

Thus,

$$a + e^{\mathbf{i}\omega\ell} x = z(b + e^{-\mathbf{i}\omega\ell} x),$$

or

$$x = \frac{zb - a}{e^{\mathbf{i}\omega\ell} - ze^{-\mathbf{i}\omega\ell}}. \tag{22}$$

Replacing x from (20) in (22), we have

$$\begin{aligned}y_{i,3} &= \left|c + 2\cos(\omega\ell)\frac{zb-a}{e^{\mathbf{i}\omega\ell} - ze^{-\mathbf{i}\omega\ell}}\right| \\ &= \left|c\frac{\cos(\omega\ell)(1 - z + \frac{2zb-2a}{c}) + \mathbf{i}\sin(\omega\ell)(1+z)}{\cos(\omega\ell)(1-z) + \mathbf{i}\sin(\omega\ell)(1+z)}\right|.\end{aligned} \tag{23}$$

Define the following known complex variables:

$$\begin{aligned}k_1 &= 1 - z + \frac{2zb - 2a}{c}; \\ k_2 &= 1 + z; \\ k_3 &= 1 - z; \\ k_4 &= y_{i,3}/|c|.\end{aligned}$$

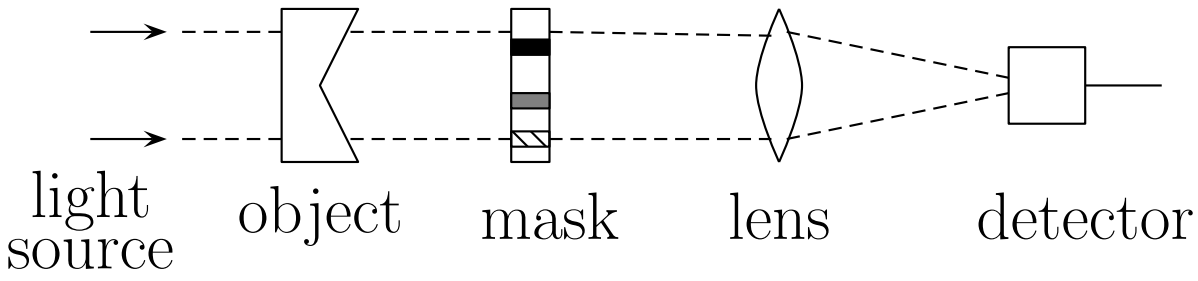


Figure 13: A typical setup for many optical system where the object of interest is passed through a coded diffraction pattern or a mask , and then through a Fourier lens.

Also let $k_1 = k_{1r} + \mathbf{i}k_{1i}$ and use similar notation for the real and imaginary parts of other variables. Then, one can square (23) to get

$$\begin{aligned}(k_{1r}\cos(\omega\ell) - k_{2i}\sin(\omega\ell))^2 + (k_{1i}\cos(\omega\ell) + k_{2r}\sin(\omega\ell))^2 \\ = k_4^2[(k_{3r}\cos(\omega\ell) - k_{2i}\sin(\omega\ell))^2 + (k_{3i}\cos(\omega\ell) + k_{2r}\sin(\omega\ell))^2].\end{aligned}$$

Now defining appropriate new known real variables k_5, k_6 and k_7, we get an equation of the form

$$k_5\cos^2(\omega\ell) + k_6\sin^2(\omega\ell) = k_7\sin(\omega\ell)\cos(\omega\ell).$$

Squaring the above equation and using $\sin^2(\omega\ell) = 1 - \cos^2(\omega\ell)$, we get a quadratic equation in $\cos^2(\omega\ell)$ that one can easily solve to find at most 2 possible values for ℓ. Note that $\cos(\omega\ell)$ is positive by construction. Now since there are two possible values of α, one can get at most 4 solutions for ℓ and x_ℓ. Those solutions can be checked by (21). If the guess is true, the probability that the check fails is 0; thus, one can recover the resolvable multiton with probability 1.

3 Fourier-Friendly Sparse Case

In some applications such as optical imaging [20, 30], the design of the measurement matrix cannot be arbitrary. In optical imaging, the object of interest, signal x, can be passed through an optical diffraction pattern or a mask and an optical Fourier lens. A typical setup for optical imaging is shown in Figure 13. With a complex-valued mask, we can modulate each component of the signal x_i by some complex number m_i, while the lens takes the Fourier transform of the signal. For example, consider passing the signal through a mask and then Fourier lens which is common in optical imaging. The output of this transform is FMx, where F is the DFT matrix of length n and $M \in \mathbb{C}^{n\times n}$ is a diagonal mask matrix (Figure 14). In general, it is possible to have multiple stages of masks and lenses. While increasing the number of stages can make the system more complex, in many optical systems, having up to two stages is considered practical [39,40]. In our proposed solution, we will have at most two masks for all measurements.

In this section, we show how one can have a Fourier-friendly implementation of the set of measurements described in previous sections for the sparse case. We first provide an overview of the result of [6] on constructing a sparse-graph-code using "Chinese Remainder Theorem", in Subsection 3.1. In Subsection 3.1, we show how our proposed measurements can be obtained in a Fourier-friendly setup, with the aid of the result of [6].

3.1 Ensemble of Graphs Constructed by Chinese Remainder Theorem

In this subsection, we provide a brief overview of the result in [6] that uses the "Chinese Remainder Theorem" (CRT) to construct a deterministic and well-structured coding matrix that is also of practical interest.

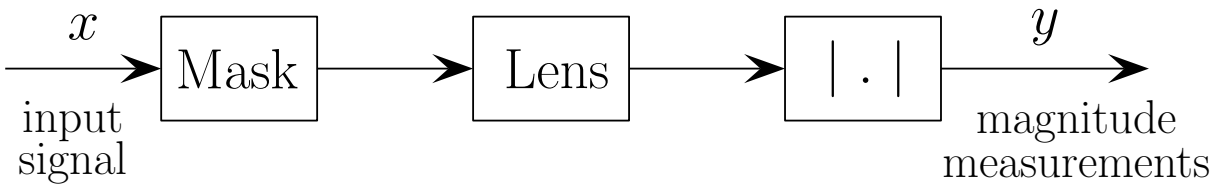


Figure 14: The block diagram of an optical imaging system where signal x is passed through a mask (modulated by a diagonal matrix), and then passed through a lens (DFT matrix). The magnitude block, $|.|$, is showing that the phase information is not available in the measurements.

We use this construction to design a Fourier-friendly measurement matrix for the sparse case. For more details about the theory of ensemble of graphs constructed by the CRT, we refer the readers to [6].

In Section 2, we analyzed the performance of Unicolor PhaseCode for the ensemble of graphs $\mathcal{C}_1^K(d, m)$. In this ensemble which is based on the balls and bins model, each ball goes to exactly d bins uniformly at random. Now we consider another ensemble $\mathcal{C}_2^K(\mathcal{F}, m)$ based on balls and bins model. Define the set $\mathcal{F}$ as $\mathcal{F} = \{f_1, f_2, \ldots, f_d\}$. Partition the bins into d sets. Let the number of bins in stage i be f_i; thus, $\sum_{i=1}^{d} f_i = m$. In this construction, each ball goes to exactly one bin per stage. Therefore, we again end up with having a bipartite graph with left regular degree equal to d. Assuming that $f_i = F + \mathcal{O}(1)$ for all i and consequently $F = \mathcal{O}(K)$, the edge degree distribution of the right nodes does not change for large enough K and is given in (9). Therefore, the tree analysis and the density evolution equation stated in (10) remain the same, and one can essentially get all the previous results using this ensemble.

Note that sampling a graph from $\mathcal{C}_2^K(\mathcal{F}, m)$ has no practical advantage over sampling from the ensemble $\mathcal{C}_1^K(d, m)$. However, we use the CRT to show that if the K non-zero components of the signal is chosen uniformly at random with replacement from the n components, and if K is in the sub-linear regime, that is $\frac{K}{n} \to 0$, one can design a deterministic coding matrix which consists of d stages of sub-matrices with rows that are circularly-shifted versions of a deterministic *subsampling* pattern. The subsampling rate at stage i is f_i. In the following example we demonstrate how the deterministic matrix is constructed.

Example Suppose that the coding matrix has two stages with $f_1 = 2$ and $f_2 = 3$. Assume that $n = 6$. Then, the coding matrix is

$$\left(\begin{array}{cccccc} 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ \hline 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 \end{array}\right)$$

Now, we formally define the ensemble of graphs constructed by the CRT. First, set $n = \prod_{i=1}^{d} f_i$. Partition the set of $m = \sum_{i=1}^{d} f_i$ right nodes to d stages in the trivial way. Suppose that the K non-zero components of signal are chosen uniformly at random with replacement from the n components, and K is in the sub-linear regime, that is $\frac{K}{n} \to 0$. Note that the "with replacement" assumption might lead to having a signal with less than K non-zero components, but this is only a technical assumption that we need to make, and via simulations we will show the good performance of the CRT-based code for exactly K-sparse signal. Let $I = (i_1, i_2, \ldots, i_K)$ denote the non-zero components where $1 \leq i_k \leq n,\ 1 \leq k \leq K$. We associate the integers from 0 to $n-1$ to d numbers $(r_1, r_2, \ldots, r_d)$ using the CRT, where $0 \leq r_i \leq f_i - 1$; thus, i_k uniquely determines one bin per stage. How this association is done will be explained shortly. Then, each active left node i_k is connected to the associated set of bins that are determined by $(r_1, r_2, \ldots, r_d)$. The ensemble $\mathcal{C}_3^K(\mathcal{F}, m)$ is the collection of all the graphs that are constructed as described. Furthermore, the

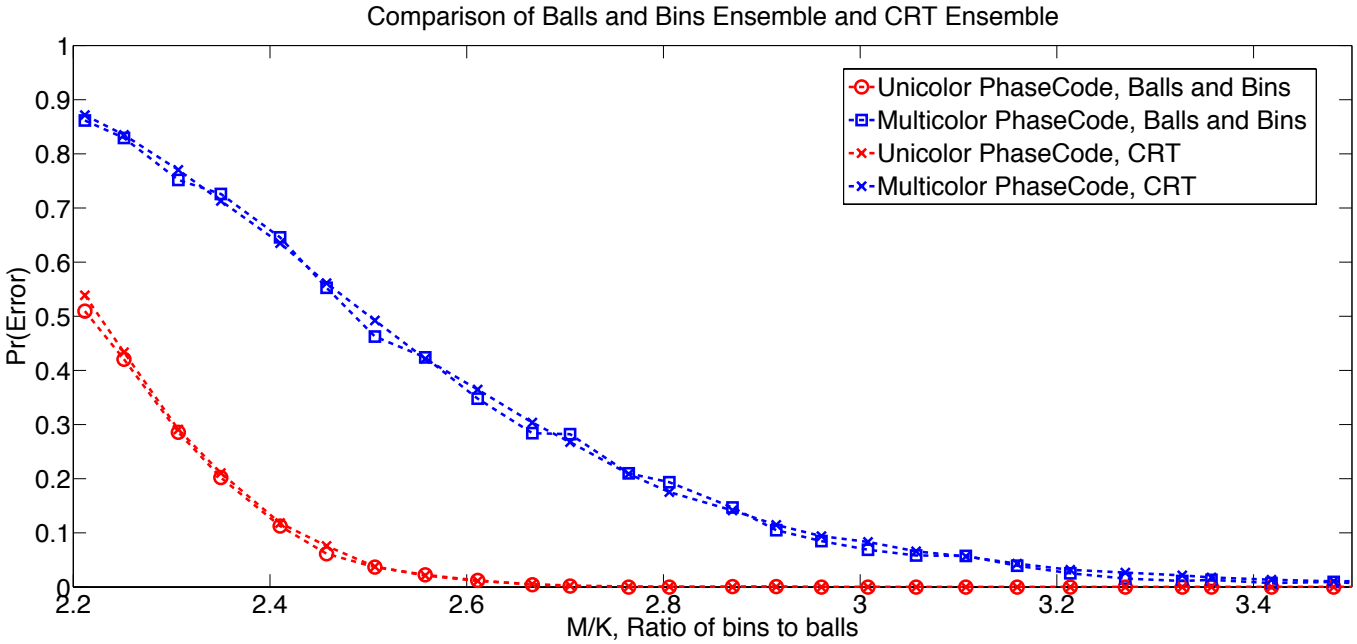


Figure 15: **Comparison of Balls and Bins Ensemble and CRT Ensemble.** We evaluate Unicolor and Multicolor PhaseCode algorithms with balls and bins ensembles and CRT ensembles. We chose the left degree $d = 7$, and constructed an appropriate CRT ensemble based on $\mathcal{F} = \{47, 49, 50, 53, 57, 59, 61\}$. The number of bins is determined by $\mathcal{F}$, i.e., $M = \sum_{i=1}^{d} f_i = 376$. By varying K from 107 to 170, we effectively control M/K, the ratio of bins to balls. Then, we also evaluate performance of algorithms with Balls and Bins ensembles, and compare results. Each point is averaged over 10000 runs. We observe a negligible difference between algorithms' performance with Balls and Bins ensembles and performance with CRT ensembles.

uniformly random selection of I makes sure that all these graphs occur with equal probability. See [6] for details.

To show how we associate I to $(r_1, r_2, \ldots, r_d)$, we need to review the Chinese Remainder Theorem. Let $n = \prod_{i=1}^{d} n_i$ and n_i's are pairwise co-prime positive integers. The theorem states that every integer n' between 0 and $n-1$ is uniquely represented by the sequence $(r_1, r_2, ..., r_d)$ of its remainders modulo $n_1, n_2, \ldots, n_d$ respectively and vice-versa. We use this unique CRT mapping to associate the active left nodes with d right nodes.

Lemma 3.1. *[6] The ensembles $\mathcal{C}_2^K(\mathcal{F}, m)$ and $\mathcal{C}_3^K(\mathcal{F}, m)$ are identical.*

Proof. Clearly, $\mathcal{C}_3^K(\mathcal{F}, m) \subset \mathcal{C}_2^K(\mathcal{F}, m)$. The reverse is also true by CRT since there is a unique integer between 0 to $n-1$ with remainders r_i modulo f_i for all i. ■

Figure 15 evaluates the performance of PhaseCode Algorithms with two ensembles: $\mathcal{C}_1^K(d, m)$ and $\mathcal{C}_3^K(\mathcal{F}, m)$. We chose $d = 7$ and $\mathcal{F} = \{47, 49, 50, 53, 57, 59, 61\}$. Thus, $M = \sum_{i=1}^{d} f_i = 376$. We varied the value of K ($107 \leq K \leq 170$) such that M/K varies between 2.2 and 3.5. Each point is averaged over 10000 runs to determine the error probability. One can observe a negligible difference between performance of the algorithms with the balls and bins ensemble and with the CRT ensemble.

In the following we provide remarks of how one can extend the above construction of CRT.

Remark In the above example of CRT construction, we implicitly assumed $K = \mathcal{O}(n^{1/d})$. The technique can be simply extended to cases where $K = \mathcal{O}(n^{\alpha/d}), \forall\ i$ by using taller stages. Instead of using $\mathcal{F}$ as

heights of d stages, we use $\mathcal{F}' = \{f'_1, ..., f'_d\}$ as heights, where

$$f'_i = \prod_{j=0}^{\alpha-1} f_{((i+j) \bmod d)+1}.$$

For example, if $\alpha = 2$ and $d = 7$, one can convert a set of coprimes $\mathcal{F}$ into a set of heights $\mathcal{F} = \{f_1 f_2, f_2 f_3, f_3 f_4, f_4 f_5, f_5 f_6, f_6 f_7, f_7 f_1\}$. Then, $m = \sum_{i=1}^{d} \prod_{j=0}^{\alpha-1} n_{((i+j) \bmod d)+1} = \mathcal{O}(n^{2/d})$, which is in the order of K. Because $\mathcal{F}$ can be chosen from a dense set of coprimes, one can always choose it carefully to induce a right number of measurements. For the most general case where $K = \mathcal{O}(n^{p/q})$ and $0 \leq p/q < 1$, one can use a similar extension and construction by finding q coprimes and stacking p of them in each stage. We omit details of the technique and refer interested readers to [6].

3.2 Fourier-Friendly Compressive Phase Retrieval

Without loss of generality, we consider only a 1-D case for x here, though our arguments extend in a straightforward way to 2-D images as well. Let $X = Fx$ be the Fourier transform of the signal. In Subsection 3.1, we showed that the coding matrix H can be realized using d stages of circulant matrices without changing the performance of sparse-graph-codes. To have a Fourier-friendly implementation of the CRT code matrix, we expand each stage of the $f_i \times n$ matrix to a circulant $n \times n$ matrix. Let C denote this circulant coding matrix for one stage. In the following, we show that how using our proposed CRT code matrix, one can have access to all the necessary measurements using *only diagonal masks and lenses*. Note that there are two types of measurements that we are interested in. The first type is the measurement obtained by modulating the coding matrix with complex exponentials such as $e^{\mathbf{i}\omega\ell}$. The second type involves modulating by $\cos(\omega\ell)$. First let us see how the main measurements without these modulations can be obtained if the coding matrix is circulant. The main measurements are $|\sum_j C_{ij} X_j|$. Since C is circulant, the eigenvectors of C are the columns of a unitary Fourier matrix [37]. Thus, the eigenvalue decomposition of C is $C = FMF^{-1}$ for some diagonal matrix M. Hence, we construct our measurements by modulating the signal x with the diagonal mask M and then taking a Fourier transform by using an optical lens:

$$\begin{aligned} |FMx| &= |FF^{-1}CFx| \\ &= |CFx| \\ &= |CX|. \end{aligned}$$

For each stage of the CRT code matrix (there are d stages overall), we need one physical experiment. The physical experiment corresponding to the i-th stage, where $1 \leq i \leq d$, gives us n/f_i replicas of f_i unique measurements in one shot. As illustrated in Figure 16, for each experiment, the camera measures only one copy of the f_i measurements. Let $y_i \in \mathcal{C}^{f_i}$ be the measurements corresponding to stage i. Then, the measurements of the different stages are gathered to form the measurement vector $y \in \mathcal{C}^m$ as follows:

$$y = [y_1^T, y_2^T, \ldots, y_d^T]^T.$$

Thus, the actual sample complexity is still $m = \sum_{i=1}^{d} f_i = \mathcal{O}(K)$.

Now we explain how one can get access to all the necessary measurement $y_{1,i}$ to $y_{4,i}$. The measurements corresponding to $y_{1,i}$, $y_{2,i}$ and $y_{4,i}$ involve the signal modulated by some pure phase $e^{\mathbf{i}\phi}$ in the Fourier domain which corresponds to a circularly shifted x in the time domain. To get $y_{1,i}$, it is enough to pass a circularly shifted $x(i+1)$ through a mask and Fourier transform, since $FSx = [X(\ell)e^{\mathbf{i}\omega\ell}]$, where S is a forward shift-by-1 matrix. Similarly, one can get $y_{2,i}$ by using $x(i-1)$ and shifting the signal backwards. The check measurement $y_{4,i}$ can also be obtained by shifting x randomly and passing it through a mask and Fourier transform.

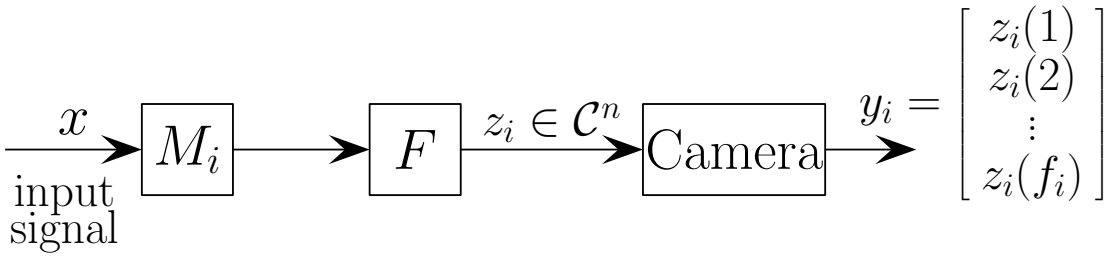


Figure 16: The block diagram of Fourier-friendly compressive phase retrieval using the CRT matrix. The figure shows stage i of the CRT matrix ($1 \leq i \leq d$). The signal of interest, x, is passed through a binary mask corresponding to stage i, and then the Fourier lens. The output of this experiment is signal z_i of length n. However, these n measurements are not unique; they are n/f_i replicas of f_i unique measurements. Thus, the camera only reads the first f_i components of z_i.

Finally, we realize measurements $y_{3,i}$ by using 3 blocks of Fourier transforms (lenses) and 2 masks as follows.[12] Let $\tilde{D}$ be a diagonal matrix such that $\tilde{d}_{\ell\ell} = \cos(\omega\ell)$. We are interested in constructing the measurements of the form $|C\tilde{D}X|$. This can be done by using two masks, $\tilde{D}$ and M, with three Fourier lenses as follows.

$$|FMF\tilde{D}Fx| = |FFCF^{-1}F\tilde{D}X| \tag{24}$$

$$= |F^2C\tilde{D}X|. \tag{25}$$

Note that F^2 is just a permutation matrix so we can construct all the measurements $y_{3,i}$ using only real masks and Fourier lenses.

Remark So far, all of our proposed masks are *real* masks that are of great practical interest due to their ease of implementation. It is clear that $\tilde{D}$ is real. Furthermore, the DFT of an impulse train or a sub-sampling pattern is a binary vector [37]. Thus, M is also a binary mask.

Remark In some optical settings, a circulant shift of the signal might be hard to implement. In this case, our three sets of measurement, $y_{1,i}$, $y_{2,i}$ and $y_{4,i}$, can be realized similarly to $y_{3,i}$, this time by using a complex diagonal mask. For example, to realize $y_{1,i}$, we replace $\tilde{D}$ in (24) by $D' = \text{diag}[e^{\mathbf{i}\omega\ell}]$.

4 The Non-Sparse Case

In this section, we use similar ideas as in [3] to propose $3n - 2$ carefully chosen measurements that enable us to recover the signal with high probability under some mild assumptions. We consider 3 measurement matrices: one of size $n \times n$ and two of size $(n-1) \times n$. The first one is the identity matrix that gives us access to all the magnitudes of the different components of x. The second measurement matrix is

$$A_1 = \begin{pmatrix} 1 & 1 & 0 & \dots & & 0 \\ 1 & 0 & 1 & 0 & \dots & 0 \\ 1 & 0 & 0 & 1 & \dots & 0 \\ & & & \vdots & & \\ 1 & 0 & \dots & & 0 & 1 \end{pmatrix}$$

[12] Another way to get measurements $y_{3,i}$ is by adding two circularly shifted signals in the time domain and passing it through a mask and Fourier transform. Note that the Fourier transform of $x(i+1) + x(i-1)$ is $2X(\ell)\cos(\omega\ell)$. However, this approach is less practical in some optical systems.

Specifically, for $1 \leq j, \ell \leq n$, $A_1(j,\ell) = 1_{\{\ell=1\}} + 1_{\{j=\ell-1\}}$. Thus, we have access to measurements of the form $|x_1 + x_\ell|,\ 2 \leq \ell \leq n$. Let

$$D = \text{diag}(e^{\mathbf{i}\omega(\ell-1)}) = \begin{bmatrix} 1 & 0 & \dots & 0 \\ 0 & e^{\mathbf{i}\omega} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & e^{\mathbf{i}\omega(n-1)} \end{bmatrix},$$

that is $D_{j\ell} = e^{\mathbf{i}\omega(\ell-1)} 1_{\{\ell=j\}}$. The third measurement matrix is $A_1 D$. Then, we have access to measurements of the form $|x_1 + e^{\mathbf{i}\omega(\ell-1)} x_\ell|,\ 1 \leq \ell \leq n-1$.

Theorem 4.1. *Suppose that $x_1 \neq 0$. Then x can be exactly recovered from the $3n-2$ measurements $|Ix|$, $|A_1 x|$, and $|A_1 D x|$ up to a global phase.*

Proof. We prove the theorem by explicitly showing how x can be recovered from the $3n-2$ measurements. Without loss of generality, set the phase of x_1 to be 0. Define the following known variables.

$$\begin{aligned} y_1 &= |x_1|; \\ y_2 &= |x_2|; \\ y_3 &= |x_1 + x_2|; \\ y_4 &= |x_1 + e^{\mathbf{i}\omega} x_2|. \end{aligned}$$

Let $x_2 = |x_2| e^{\mathbf{i}\phi}$. Then, by the cosine law we have

$$\cos(\phi) = \frac{y_3^2 - y_1^2 - y_2^2}{2 y_1 y_2} = \alpha.$$

Assuming that the well-defined function $\cos^{-1}(\alpha)$ for $-1 \leq \alpha \leq 1$ returns values between 0 and π, we know that $\phi = \pm \cos^{-1}(\alpha)$. Thus, we need to resolve the ambiguity of the sign. Clearly, there is no ambiguity if $\phi = 0$ or $\phi = \pi$. We show that one can use y_4 to find the sign. Note that

$$y_4^2 = y_1^2 + y_2^2 + 2 y_1 y_2 \cos(\phi + \omega).$$

Let $\phi_1 = \cos^{-1}(\alpha)$ and $\phi_2 = -\cos^{-1}(\alpha)$. Then, exactly one of the following equalities will be true, which resolves the phase of x_2:

$$\cos(\phi_1 + \omega) = \frac{y_3^2 - y_1^2 - y_2^2}{2 y_1 y_2},$$

or,

$$\cos(\phi_2 + \omega) = \frac{y_4^2 - y_1^2 - y_2^2}{2 y_1 y_2}.$$

Note that the value of ω is known. Thus, x_2 is now completely resolved with respect to x_1. Similarly, $x_\ell,\ 2 \leq \ell \leq n$ can be resolved with respect to x_1; thus, x can be uniquely recovered. ■

Remark Note that x_1 could have been replaced, without loss of generality, with any other $x_j, 2 \leq j \leq n$.

Remark Note that the set of measurements that we consider is not *injective* in the sense of [4], since if x_1 is zero, the set of measurements can be mapped to different signals x. Thus, we are not violating the conjecture in [7] that $4n-4$ measurements is the fundamental limit to have an injective map. However, we consider the event of one signal component being exactly 0 as a measure-0 event in the non-sparse case.

4.1 Fourier-Friendly Non-Sparse Case

For the case that x is not sparse, the set of measurements proposed in Section 4 can also be implemented using a diagonal mask and a Fourier lens. Consider the 3 measurement matrices (I, A_1, A_1D). Towards this end, we use a similar trick to [3]. The trick is to find all the signal components in Fourier domain $X_i,\ 1 \leq i \leq n$ with respect to the first component in time domain that is x_1.[13] We assume that $x_1 \neq 0$. Consider the following three diagonal mask matrices: $M_1 = I = \text{diag}([1, 1, \ldots, 1])$, $M_2 = \text{diag}([2, 1, \ldots, 1])$, and $M_3 = \text{diag}([1 + \mathbf{i}, 1, \ldots, 1])$. The sample complexity of the Fourier-friendly algorithm is $3n$, while the number of physical experiments to be done is 3. Clearly, $|FM_1x|$ gives measurements $|X_i|,\ 1 \leq i \leq n$. On the other hand, we have

$$|FM_2x| = |X + x_1[1, 1, \ldots, 1]^T|,$$
$$|FM_3x| = |X + x_1 \times \mathbf{i}[1, 1, \ldots, 1]^T|.$$

First we show that we can find $|x_1|$ using the above measurements. Define the following known variables:

$$y_1 = |X_1|$$
$$y_2 = |X_1 + x_1|$$
$$y_3 = |X_1 + \mathbf{i}x_1|.$$

Then, after some algebra one finds the equation

$$(y_2^2 - y_1^2 - |x_1|^2)^2 + (y_3^2 - y_1^2 - |x_1|^2)^2 = 4y_1^2|x_1|^2.$$

Note that the above equation has at most two solutions for x_1. Moreover, we can get n equations similar to the one above using $|X_i|$, $|X_i + x_1|$ and $|X_i + \mathbf{i}x_1|$ which assures that $|x_1|$ can be uniquely found with high probability. Now, since x_1 is non-zero by assumption, one sets the phase of x_1 to zero without loss of generality, and find all the X_i's relative to x_1 using the procedure described in Section 4.

In this case we are interested in chaining the components of the signal in Fourier domain, that is finding measurements $|X|$, $|A_1X|$ and $|A_1DX|$ where $X = Fx$. Similar to Section 4, this time one needs to assume that X_i is non-zero for all i. Finding $|Fx|$ is of course trivial using an identity mask. For the other measurements, note that A_1 becomes a circulant matrix if one adds the row $[1, 0, \ldots, 0, 1]$ to the end of the matrix. Therefore, one can write

$$\begin{aligned} |A_1X| &= |F\Lambda F^{-1}Fx| \\ &= |F\Lambda x|, \end{aligned}$$

for some diagonal mask Λ. To realize the third measurement, recall that a shift in time domain corresponds to multiplying by complex exponential in the Fourier domain. Thus, considering the forward shift matrix S, one has

$$\begin{aligned} |F\Lambda Sx| &= |F\Lambda F^{-1}FSx| \\ &= |A_1DX|. \end{aligned}$$

As one can see, all the necessary measurements can be realized by diagonal masks and Fourier lenses.

For the second set of measurements, we use a similar trick to [3]. This time, we find all the signal components in Fourier domain $X_i,\ 1 \leq i \leq n$ with respect to the first component in time domain that is x_1. Consider the following three diagonal mask matrices: $M_1 = I = \text{diag}([1, 1, \ldots, 1])$, $M_2 = \text{diag}([2, 1, \ldots, 1])$,

[13] Note that x_1 could have been replaced by any $x_j,\ 2 \leq j \leq n$.

and $M_3 = \text{diag}([1+\mathbf{i}, 1, \ldots, 1])$. Then, clearly $|FM_1x|$ gives measurements $|X_i|,\ 1 \leq i \leq n$. On the other hand, we have

$$|FM_2x| = |X + x_1[1, 1, \ldots., 1]^T|$$
$$|FM_3x| = |X + x_1 \times \mathbf{i}[1, 1, \ldots., 1]^T|.$$

First we show that we can find $|x_1|$ using the above measurements. Define the following known variables:

$$y_1 = |X_1|$$
$$y_2 = |X_1 + x_1|$$
$$y_3 = |X_1 + \mathbf{i}x_1|.$$

Then, after some algebra one finds the equation

$$(y_2^2 - y_1^2 - |x_1|^2)^2 + (y_3^2 - y_1^2 - |x_1|^2)^2 = 4y_1^2|x_1|^2.$$

Note that the above equation has at most two solutions for x_1. Moreover, we can get n equations similar to the one above using $|X_i|$, $|X_i + x_1|$ and $|X_i + \mathbf{i}x_1|$ which assures that $|x_1|$ can be uniquely found with high probability. Now, assuming x_1 is non-zero, one sets the phase of x_1 to zero without loss of generality, and find all the X_i's relative to x_1 using the procedure described in Section 4.

5 Conclusion and Future Work

We considered the problem of recovering a complex signal $x \in \mathbb{C}^n$ from m intensity measurements of the form $|a_ix|,\ 1 \leq i \leq m$, where a_i is a measurement row vector. We addressed multiple settings corresponding to whether the measurement vectors are unconstrained choices or not, and to whether the signal to be recovered is sparse or not.

Our main focus was on the case where the measurement vectors are unconstrained, and where x is exactly K-sparse, or the so-called general compressive phase-retrieval problem. We proposed Unicolor PhaseCode and Multicolor PhaseCode algorithms that are based on a sparse-graph-codes framework. We showed that Unicolor PhaseCode can provably recover all but a tiny fraction of the non-zero signal components with high probability. Towards this end, we used coding-theoretic tools like density evolution for the design and analysis of Unicolor PhaseCode. This contrasts and complements popular approaches to the phase retrieval problem based on alternating-minimization, convex-relaxation, and semi-definite programming. To the best of our knowledge, our work is the first one that characterizes the precise number of measurements needed to guarantee high reliability, rather than only Big Oh statements.

Our proposed algorithms have both an order-optimal $O(K)$ decoding time and an order-optimal $O(K)$ memory complexity. We also showed that the proposed algorithms can be used for practical systems such as optical systems with proper modifications. Via extensive simulation results, we validated the performance of our proposed algorithms both in unconstrained settings and constrained settings.

For the general non-sparse signal case, we proposed a simple but efficient deterministic set of $3n - 2$ measurements, and a decoding algorithm that can recover the signal of interest, assuming that a particular component of the signal is non-zero. We also showed that a variant of the proposed scheme can be used in practical optical systems.

Various sources of noise must be considered in practical systems: a K-sparse signal can be 'approximately K-sparse; measurements can be corrupted by noise; masks, through which the signal is passed, might be inaccurate. Our proposed PhaseCode algorithms can be robustified, while retaining the basic sparse-graph-code architecture of the underlying noiseless PhaseCode algorithm. That is, in the robust PhaseCode algorithm, the code graph matrix H remains as in the noiseless setting, but the four trigonometric measurements comprising the G modulation matrix are increased to $O(\log n)$ appropriately chosen

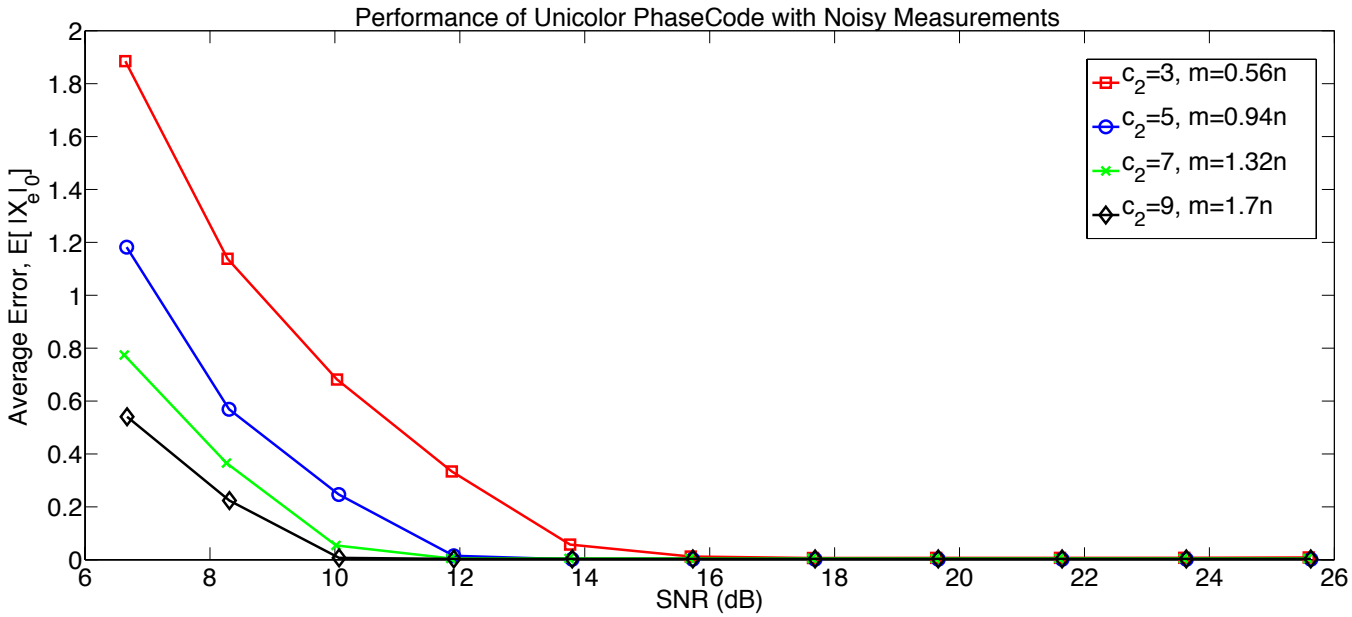


Figure 17: **Performance of PhaseCode with Noisy Measurements.** We evaluate the robustified Unicolor PhaseCode algorithm with noisy measurements, i.e., $y = |Ax| + w$. We assumed additive white Gaussian noise for simplicity. Signal-to-noise ratio, SNR, is defined as the power ratio between a measured signal and the added noise signal. We assumed a finite number of constellation points that each symbol of x can take, and measured the L_0 norm of estimate errors, $\mathbb{E}[\|x_e\|_0]$. We used a random signal of length $n = 2048$ and sparsity $K = 5$. We varied the number of measurements from $0.56n$ to $1.7n$, and varied SNR. Each point is averaged over 500 runs. The robustified Unicolor PhaseCode algorithm is observed to perform well even under the absence of noise.

measurements. This expansion is needed to deal with noise while reliably performing the requisite guess-and-check tests in a robust manner. Without providing much details about the actual measurement designs in this paper, we provide simulation results with noisy measurements in Figure 17, which shows how well our robust PhaseCode algorithm performs in the presence of noise. Note that for the noisy simulations, each element of the measurement matrix H is drawn independently according to a Bernoulli distribution. This is in contrast to the fixed left-degree design described in Section 2.3.

We conclude our paper by presenting interesting directions to pursue from this point. Here, we provide a few of these direction.

(i) It is interesting to investigate the PhaseCode algorithm in the presence of noise, from a theoretical point of view. To the best of our knowledge, there is no strong theoretical guarantee for any of the popular approaches in the presence of noise; though, some robustness results have been shown through simulations. The power techniques of coding theory can be helpful to provide theoretical guarantees for the performance of PhaseCode algorithm in the presence of noise.

(ii) A future direction is to come up with a rigorous analysis of MultiColor PhaseCode. A more careful study of the Multicolor PhaseCode (which is beyond the scope of this paper), reveals that the Multi-Color PhaseCode algorithm corresponds to a peculiar but interesting stochastic process on the sparse graph. The study of this process can be of great theoretical interest.

(iii) It is interesting to see whether other ideas from coding theory can be used to design an algorithm that provides higher reliability, or requires fewer number of measurements. Of course, the ultimate goal is to find a low-complexity capacity-achieving algorithm , i.e. an algorithm that requires only $4K - O(1)$ measurements and has a probability of success that goes to 1 as K goes to infinity.

References

[1] B. Bollobas, "Random graphs," *Cambridge University Press*, 2001.

[2] S. Cai, M. Bakshi, S. Jaggi, and M. Chen, "SUPER: Sparse signals with Unknown Phases Efficiently Recovered," *arXiv preprint arXiv:1401.4451*, 2014.

[3] V. Pohl, F. Yang, and H. Boche, "Phase retrieval from low rate samples," *arXiv preprint arXiv:1311.7045*, 2013.

[4] R. Balan, P. G. Casazza, and D. Edidin, "On signal reconstruction without phase," *Applied and Computational Harmonic Analysis*, vol. 20, no. 3, May 2009.

[5] 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, Feb. 2001.

[6] S. Pawar and K. Ramchandran, "Computing a k-sparse n-length Discrete Fourier Transform using at most $4k$ samples and $O(k \log k)$ complexity," *arXiv preprint arXiv:1305.0870*, 2013.

[7] A. S. Bandeira, J. Cahill, D. G. Mixon, and A. A. Nelson, "Fundamental Limits of Phase Retrieval," *Proc. 10th Intern. Conf. on Sampling Theory and Applications (SampTA)*, July 2013.

[8] R. P. Milane, "Phase retrieval in crystallography and optics," *J. Opt. Soc. Am. A*, vol. 7, pp. 394–411, 1990.

[9] R. W. Harrison "Phase problem in crystallography," *JOSA A*, vol. 10, pp. 1046–1055, 1993.

[10] M. H. Hayes, J. S. Lim, and A. V. Oppenheim, "Signal reconstruction from phase or magnitude," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 28, no. 6, pp. 672–680, 1980.

[11] O. Bunk, A. Diaz, F. Pfeiffer, C. David, B. Schmitt, D. K. Satapathy, and J. F. Veen, "Diffractive imaging for periodic samples: retrieving one-dimensional concentration profiles across microfluidic channels," *Acta Crystallographica Section A: Foundations of Crystallography*, vol. 63, no. 4, pp. 306–314, 2007.

[12] J. C. Dainty and J. R. Fienup, "Phase retrieval and image reconstruction for astronomy," *Image Recovery: Theory and Application, Academic Press*, pp. 231–275, 1987.

[13] 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.

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

[15] 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.

[16] E. J. Candes, Y. C. Eldar, T. Strohmer, and V. Voroninski, "Phase retrieval via matrix completion," *SIAM Journal on Imaging Sciences*, vol. 6, no. 1, pp. 199–225, 2013.

[17] E. J. Candes, X. Li, and M. Soltanolkotabi, "Phase retrieval from coded diffraction patterns," *arXiv preprint arXiv:1310.3240*, 2013.

[18] E. J. Candes, J. Romberg, and T. Tao, "Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information," *IEEE Trans. Inform. Theory*, vol. 52, no. 2, pp. 489–509, 2006.

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

[20] J. M. Rodenburg, "Ptychography and related diffractive imaging methods," *Advances in Imaging and Electron Physics*, vol. 150, pp. 87–184, 2008.

[21] K. Jaganathan, S. Oymak, and B. Hassibi, "Sparse phase retrieval: Convex algorithms and limitations," *Proceedings of IEEE International Symposium on Information Theory (ISIT)*, pp. 1022–1026, 2013.

[22] K. Jaganathan, S. Oymak, and B. Hassibi, "Phase retrieval for sparse signals using rank minimization," *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 3449–3452, 2012.

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

[24] D. Donoho, "Compressed sensing," *IEEE Trans. Inform. Theory*, vol. 52, no. 4, 2006.

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

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

[27] B. Alexeev, A. S. Bandeira, M. Fickus, and D. G. Mixon, "Phase retrieval with polarization," *arXiv preprint arXiv:1210.7752*, 2012.

[28] A. Walther, "The question of phase retrieval in optics," *Optica Acta*, vol. 10, no. 1, pp. 41–49, 1963.

[29] M. Mirhosseini, O. S. Magana-Loaiza, S. M. Hashemi Rafsanjani, R. W. Boyd, "Compressive direct measurement of the quantum wavefunction," *arXiv preprint arXiv:1404.2680*, 2014.

[30] E. G. Loewen and E. Popov, "Diffraction gratings and applications," *CRC Press*, 1997.

[31] B. G. Bodmann and N. Hammen, "Stable phase retrieval with low-redundancy frames," *arXiv preprint arXiv:1302.5487*, 2013.

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

[33] I. Waldspurger, A. d'Aspremont, and S. Mallat, "Phase recovery, maxcut and complex semidenite programming," *Mathematical Programming*, pp. 1–35, 2013.

[34] M. Luby, "LT codes," *Proc. IEEE Foundations of Computer Science (FOCS)*, 2002.

[35] 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.

[36] Erdos, P. and Renyi, A. "On the evolution of random graphs," *Publications of the Mathematical Institute of the Hungarian Academy of Sciences*, vol. 5, pp. 17–61, 1960.

[37] A. V. Oppenheim, R. W. Schafer and J. R. Buck, "Discrete-Time Signal Processing," *Prentice Hall*, 1989.

[38] T. Richardson and R. Urbanke, "Modern Coding Theory," *Cambridge University Press*, 2008.

[39] Z. Wang, L. Millet, M. Mir, H. Ding, S. Unarunotai, J. Rogers, M. U. Gillette, and G. Popescu, "Spatial light interference microscopy (SLIM)," *Opt. Express*, vol. 19, no. 2, pp. 1016–1026, 2011.

[40] S. R. P. Pavani and R. Piestun, "Three dimensional tracking of fluorescent microparticles using a photon-limited double-helix response system," *Opt. Express*, vol. 16, pp. 22048–57, 2008.

A Probability of Tree-like Neighborhood

In this section, we give a short proof of Lemma 2.6. Let C_ℓ be the number of right-nodes and V_ℓ be the number of left-nodes in $\mathcal{N}_{\vec{e}}^{2\ell}$. Since the ensemble of the graphs we consider is only left-regular, we cannot immediately use the result of [5]. Note that the degree distribution of right nodes is Poisson distribution with constant rate. The key idea is to show that the size of the tree is bounded by $O(\log(K)^\ell)$ with high probability. This is intuitively clear since Poisson distribution has a tail decaying faster than exponential decay. To formally show this, we keep unfolding the tree up to level ℓ^*, and at each level ℓ we upper bound the probability that the size of the tree grows larger than $O(\log(K)^\ell)$. Fix some constant c_1. We upper bound the probability of not having a tree as follows.

$$\begin{aligned}\mathbb{P}(\mathcal{N}_{\vec{e}}^{2\ell^*} \text{ is not a tree}) \leq\ & \mathbb{P}(V_{\ell^*} > c_1 \log(K)^{\ell^*}) + \mathbb{P}(C_{\ell^*} > c_1 \log(K)^{\ell^*}) + \\ & \mathbb{P}(\mathcal{N}_{\vec{e}}^{2\ell^*} \text{ is not a tree} | V_{\ell^*} < c_1 \log(K)^{\ell^*}, C_{\ell^*} < c_1 \log(K)^{\ell^*}).\end{aligned}$$

Note that since the left degree is a constant, d, if V_{ℓ^*} is of order $\log(K)^{\ell^*}$ or less, C_{ℓ^*} is also of order $\log(K)^{\ell^*}$ or less. Let $\alpha_\ell = \mathbb{P}(V_\ell > c_1 \log(K)^\ell)$. Then,

$$\begin{aligned}\alpha_\ell &\leq \alpha_{\ell-1} + \mathbb{P}(V_\ell > c_1 \log(K)^\ell | V_{\ell-1} < c_1 \log(K)^{\ell-1}) && (26)\\ &\leq \alpha_{\ell-1} + \mathbb{P}(V_\ell > c_1 \log(K)^\ell | C_\ell < c_2 \log(K)^{\ell-1}), && (27)\end{aligned}$$

where (27) is due to the fact that every left node has exactly d edges connected to right nodes so if $V_{\ell-1} < c_1 \log(K)^{\ell-1}$, there exists some constant c_2 such that $C_\ell < c_2 \log(K)^{\ell-1}$. To count the number of left nodes in depth ℓ, let $n_\ell < C_\ell$ be the number of right nodes exactly at depth ℓ after unfolding the tree. Let $X_i,\ 1 \leq i \leq n_\ell$ be the degree of these right nodes. Given that $V_{\ell-1} < c_1 \log(K)^{\ell-1}$, one has $V_\ell > c_1 \log(K)^\ell$, only if $X = \sum_{i=1}^{n_\ell} X_i > c_3 \log(K)^\ell$ for some constant c_3. The distribution of X is Poisson distribution with parameter $n_\ell \lambda$. We know that the tail probability of a Poisson random Y variable with parameter λ can be upper bounded as follows: $\mathbb{P}(Y \geq y) \leq \left(\frac{e\lambda}{y}\right)^y$. Thus,

$$\mathbb{P}(X > c_3 \log(K)^\ell) \leq \left(\frac{c_4}{\log(K)}\right)^{c_3 \log(K)^\ell} \leq O(\frac{1}{K}).$$

Thus,

$$\alpha_\ell \leq \alpha_{\ell-1} + \frac{c_5}{K}, \tag{28}$$

for some constant c_5. Now since ℓ^* is a constant, summing up the inequalities in (28), we show that

$$\alpha_{\ell^*} = \mathbb{P}(V_{\ell^*} > c_1 \log(K)^{\ell^*}) \leq O(\frac{1}{K}).$$

Similarly one can show that

$$\mathbb{P}(C_{\ell^*} > c_1 \log(K)^{\ell^*}) \leq O(\frac{1}{K}).$$

To complete the proof, we need to show that with high probability, we have a tree-like neighborhood, given that the number of nodes is bounded by order of $\log(K)^{\ell^*}$. First, we find a lower bound on the probability that $\mathcal{N}_{\vec{e}}^{2\ell+1}$ is a tree-like neighborhood if $\mathcal{N}_{\vec{e}}^{2\ell}$ is a tree-like neighborhood, when $\ell < \ell^*$. Assume that t additional edges have been revealed at this stage without forming a cycle. The probability that the next edge from a left node does not create a cycle is the probability that it is connected to one of the bins that are not already in the subgraph which is lower bounded by $1 - \frac{C_{\ell^*}}{m}$. Thus, the probability that $\mathcal{N}_{\vec{e}}^{2\ell+1}$ is a tree-like neighborhood if $\mathcal{N}_{\vec{e}}^{2\ell}$ is a tree-like neighborhood, is lower-bounded by $(1 - \frac{C_{\ell^*}}{M})^{C_{\ell+1}-C_\ell}$. Similarly, the probability that $\mathcal{N}_{\vec{e}}^{2\ell+2}$ is a tree-like neighborhood if $\mathcal{N}_{\vec{e}}^{2\ell+1}$ is a tree-like neighborhood, is lower-bounded by $(1 - \frac{V_{\ell^*}}{K})^{V_{\ell+1}-V_\ell}$. Therefore, the probability that $\mathcal{N}_{\vec{e}}^{2\ell^*}$ is a tree-like neighborhood is lower-bounded by

$$(1 - \frac{V_{\ell^*}}{K})^{V_{\ell^*}} (1 - \frac{C_{\ell^*}}{M})^{C_{\ell^*}}.$$

For large M and K, the above expression is approximately

$$e^{-(V_{\ell^*}^2/K + C_{\ell^*}^2/M)} \geq 1 - (V_{\ell^*}^2/K + C_{\ell^*}^2/M).$$

Now since V_{ℓ^*} and C_{ℓ^*} are upper-bounded by order of $\log(K)^{\ell^*}$, the probability of having a tree-like neighborhood is at least $1 - O(\log(K)^{\ell^*}/K)$.

B Convergence to Cycle-free Case

In this section, we give a short proof of Lemma 2.7. The proof follows similar steps as in [5], with the difference that the right degree is irregular and Poisson-distributed.

First, we prove (15). Let $Z_i = 1_{\{\vec{e}_i \text{ is colored}\}}$, $1 \leq i \leq Kd$ be the indicator that $\vec{e}_i$ is colored after ℓ iterations of the algorithm. Let B be the event that $\mathcal{N}_{\vec{e}_1}^{2\ell}$ is tree-like. Then,

$$\begin{aligned}
\mathbb{E}[Z_1] &= \mathbb{E}[Z_1|B]\mathbb{P}(B) + \mathbb{E}[Z_1|\bar{B}]\mathbb{P}(\bar{B}) \\
&\leq \mathbb{E}[Z_1|B] + \mathbb{P}(\bar{B}) \\
&\leq p_\ell + \frac{\gamma \log(K)^\ell}{K},
\end{aligned}$$

for some constant γ, where the last inequality is by Lemma 2.6. Trivially, $|\mathbb{E}[Z_1|B]| \leq 1$. Furthermore, $\mathbb{E}[Z] = Kd\mathbb{E}[Z_1]$. Hence,

$$Kd(1 - \frac{\gamma \log(K)^\ell}{K}) < \mathbb{E}[Z] < Kd(p_\ell + \frac{\gamma \log(K)^\ell}{K}).$$

Then, (15) follows from choosing K large enough such that $\frac{K}{\log(K)^\ell} > \frac{2\gamma}{\epsilon}$.

Second, we prove that

$$\mathbb{P}(|Z - Kdp_\ell| > Kd\epsilon/2) < 2e^{-\beta\epsilon^2 K^{1/(2\ell+1)}}. \tag{29}$$

Then, (16) follows from (15) and (29). To prove (29), we use the standard Martingale argument and Azuma's inequality provided in [5] with some modifications to account for the right irregular degree. Suppose that we expose the Kd edges of the graph one at a time. Let $Y_i = \mathbb{E}[Z|e_1^i]$. By definition, $Y_0, Y_1, \ldots, Y_{Kd}$ is a Doob's martingale process, where $Y_0 = \mathbb{E}[Z]$ and $Y_{Kd} = Z$. To use Azuma's inequality, we find the appropriate upper bound: $|Y_{i+1} - Y_i| \leq \alpha_i$. If the right-degree is regular and equal to d_c, it is shown in [5] that α_i can be chosen as $8(d_v d_c)^\ell$. We show that when the right degree has Poisson distribution with

constant rate, the degree of all of the right nodes can be upper bounded by $O(K^{\frac{1}{2\ell+0.5}})$ with probability at least $c_6 K(e^{-\beta_1 K^{\frac{1}{2\ell+0.5}}})$ for some constants c_6 and β_1. To show this, let X be a Poisson random variable with parameter λ and c_7 be some constant. Then,

$$\mathbb{P}(X > c_7 K^{\frac{1}{2\ell+0.5}}) \leq \left(\frac{e\lambda}{c_7 K^{\frac{1}{2\ell+0.5}}}\right)^{c_7 K^{\frac{1}{2\ell+0.5}}} \leq c_6(e^{-\beta_1 K^{\frac{1}{2\ell+0.5}}}).$$

Now considering $M = O(K)$ right nodes and using union bound, one can see that the probability that all the right nodes have degree less than $O(K^{\frac{1}{2\ell+0.5}})$ is at least $1 - O(K(e^{-\beta_1 K^{\frac{1}{2\ell+0.5}}}))$. Let E be the event that at least one right node has degree larger than $c_6 K(e^{-\beta_1 K^{\frac{1}{2\ell+0.5}}})$. Given E has not happened, one can upper bound α_i^2 by $O(K^{\frac{2\ell}{2\ell+0.5}})$. Then,

$$\begin{aligned}
\mathbb{P}(|Z - Kdp_\ell| > Kd\epsilon/2) &\leq \mathbb{P}(|Z - Kdp_\ell| > Kd\epsilon/2|\bar{E}) + \mathbb{P}(E) \\
&\leq 2e^{-\frac{K^2 d^2 \epsilon^2/4}{2\sum_i \alpha_i^2}} + c_6 K(e^{-\beta_1 K^{\frac{1}{2\ell+0.5}}}) \\
&\leq 2e^{-\beta\epsilon^2 K^{1/(4\ell+1)}}.
\end{aligned}$$

C Pseudocodes

In this section, we provide pseudocode of Unicolor PhaseCode Algorithm and Multicolor PhaseCode Algorithm. In addition to them, we provide pseudocode of bin processors: singleton processor, mergeable multiton processor, and resolvable multiton processor.

Pseudocode 1 Unicolor PhaseCode Algorithm

$\mathcal{I} \leftarrow \emptyset$ ▷ No ball is found in the beginning
for each i in $\{1, 2, ..., M\}$ **do** ▷ Find all singletons
  Singleton Processor
for each i in $\{1, 2, ..., M\}$ **do** ▷ Find all doubletons and merge
  Mergeable Multiton Processor
$\text{Color}_0 \leftarrow$ Color of the largest colored component ▷ Find the largest colored component*
for each ℓ in $\mathcal{I}$ **do** ▷ Uncolor all other balls and delete all values of them
  if $\text{Color}_\ell \neq \text{Color}_0$ **then**
    $x_\ell \leftarrow$ None
    $\text{Color}_\ell \leftarrow$ None
    $\mathcal{I} \leftarrow \mathcal{I} - \{\ell\}$
while $|\mathcal{I}| < K$ and any changes are made in the previous loop **do** ▷ Keep resolving multitons
  Resolvable Multiton Processor

*One can use Breadth-first search to find the largest component of a graph and its time complexity is $\mathcal{O}(K)$.

Pseudocode 2 Multicolor PhaseCode Algorithm

$\mathcal{I} \leftarrow \emptyset$ ▷ No ball is found in the beginning
for each i in $\{1, 2, ..., M\}$ **do** ▷ Find all singletons
    Singleton Process
while $|\mathcal{I}| < k$ and any changes are made in the previous loop **do** ▷ Keep resolving multitons and merging colors
    for each i in $\{1, 2, ..., M\}$ **do**
        Resolvable Multiton Processor
        Mergeable Multiton Processor

Pseudocode 3 Singleton Processor

if $y_{i,1} = y_{i,2} = y_{i,4}$ **then** ▷ Check whether this bin is a singleton or not
    $\ell \leftarrow \frac{1}{\omega} \cos^{-1}(\frac{y_{i,3}}{2y_{i,1}})$ ▷ Find the index of the ball in this bin
    $x_\ell \leftarrow y_{i,1}$ ▷ Assign a value to the ball
    $\mathcal{I}_0 \leftarrow \mathcal{I}_0 \cup \{\ell\}$ ▷ Declare a new found ball
    $\text{Color}_\ell \leftarrow$ new color ▷ Color the new ball with a new color

Pseudocode 4 Mergeable Multiton Processor

if Bin i contains no colored ball or the number of colors in the bin is not exactly 2 **then**
    Return ▷ If this bin is not mergeable
Red, Blue $\leftarrow$ Two colors of the balls in the bin
$\mathcal{R} \leftarrow$ indices of the balls that are colored with Red
$\mathcal{B} \leftarrow$ indices of the balls that are colored with Blue
$r \leftarrow \sum_{\ell \in \mathcal{R}} x_j e^{\mathbf{i}\omega\ell}$
$b \leftarrow \sum_{\ell \in \mathcal{B}} x_\ell e^{\mathbf{i}\omega\ell}$
for each z_1 in $\{+1, -1\}$ **do** ▷ Consider two candidate
    $\phi \leftarrow z_1 \cos^{-1}\left(\frac{|r|^2+|b|^2-y_{i,1}^2}{2|r||b|}\right) + \angle r - \angle b$ ▷ Find a candidate for phase offset
    if $\left|\sum_{\ell \in \mathcal{R}} x_\ell e^{\mathbf{i}\omega'\ell} + \exp(\mathbf{i}\phi) \times \sum_{\ell \in \mathcal{B}} x_\ell e^{\mathbf{i}\omega'\ell}\right| = y_{i,4}$ **then** ▷ Check the candidate with $y_{i,4}$
        Color Red and Color Blue are combined to a new color
        for each ℓ in $\mathcal{B}$ **do** ▷ Adjust phase of balls that are colored with Color Blue *
            $x_\ell \leftarrow x_\ell \times \exp(\mathbf{i}\phi)$
        Return

*One has to color not only blue balls in this bin but all blue balls. This can be done with a special data structure based on linked-lists.

Pseudocode 5 Resolvable Multiton Processor

if Bin i contains no colored ball or balls are colored with more than 1 color **then**
  Return ▷ If this bin is not resolvable
Color $\leftarrow$ Common color of the balls
$\mathcal{I}' \leftarrow \mathcal{I} \cap \{j | H_{i,j} = 1, 1 \le j \le n\}$ ▷ Colored balls in this bin
$a \leftarrow \sum_{i \in \mathcal{I}'} x_i e^{\mathbf{i}\omega\ell}$
$b \leftarrow \sum_{i \in \mathcal{I}'} x_i e^{-\mathbf{i}\omega\ell}$
$c \leftarrow \sum_{i \in \mathcal{I}'} 2\cos(\omega\ell) x_i$
$d \leftarrow \sum_{i \in \mathcal{I}'} x_i e^{\mathbf{i}\omega'\ell}$
for each z_1 in $\{+1, -1\}$ **do** ▷ Consider two signs of α
  $\alpha \leftarrow z_1 \cos^{-1}(\frac{y_{i,3}^2 - y_{i,1}^2 - y_{i,2}^2}{2y_{i,1}y_{i,2}})$
  $z \leftarrow \frac{y_{i,1}}{y_{i,2}} \exp(\alpha\mathbf{i})$
  $k_1 \leftarrow 1 - z + \frac{2(zb-a)}{c}$
  $k_2 \leftarrow 1 + z$
  $k_3 \leftarrow 1 - z$
  $k_4 \leftarrow \frac{y_{i,3}}{|c|}$
  $k_5 \leftarrow |k_1|^2 - k_4^2 |k_3|^2$
  $k_6 \leftarrow |k_2|^2 - k_4^2 |k_2|^2$
  $k_7 \leftarrow 2\operatorname{Re}(k_1)\operatorname{Im}(k_2) - 2\operatorname{Im}(k_1)\operatorname{Re}(k_2) + k_4^2(2\operatorname{Re}(k_2)\operatorname{Im}(k_3) - 2\operatorname{Re}(k_3)\operatorname{Im}(k_2))$
  $k_8 \leftarrow k_6^2 + k_7^2 - 2k_6k_7 + k_8^2$
  $k_9 \leftarrow 2k_6k_7 - k_8^2 - 2k_7^2$
  $k_{10} \leftarrow k_7^2$
  for each z_2 in $\{+1, -1\}$ **do** ▷ Consider two solutions of a quadratic equation
    if $k_9^2 - 4k_8k_{10} < 0$ **then**
      Continue
    if $\frac{-k_9 + z_2\sqrt{k_9^2 - 4k_8k_{10}}}{2k_8} < 0$ **then**
      Continue
    $\ell' \leftarrow \cos^{-1}\left[\sqrt{\frac{-k_9 + z_2\sqrt{k_9^2 - 4k_8k_{10}}}{2k_8}}\right] / \omega$ ▷ Find a candidate of ℓ
    $x' \leftarrow \frac{zb-a}{e^{\mathbf{i}\omega\ell} - ze^{-\mathbf{i}\omega\ell}}$ ▷ Find a candidate of x_ℓ
    if $y_{i,4} = |d + e^{\mathbf{i}\omega'\ell'}x'|$ **then** ▷ Check the validity of the candidates with $y_{i,4}$
      $x_{\ell'} \leftarrow x'$ ▷ Assign a value to the ball
      $\mathcal{I}_0 \leftarrow \mathcal{I}_0 \cup \{\ell'\}$ ▷ Declare a new found ball
      $\text{Color}'_\ell \leftarrow \text{Color}$ ▷ Color the new ball with the color of the other balls in the bin
      Return