ON THE DESIGN OF DETERMINISTIC MATRICES FOR FAST RECOVERY OF FOURIER COMPRESSIBLE FUNCTIONS

J. BAILEY [*], M. A. IWEN[†], AND C. V. SPENCER[‡]

Abstract. We present a general class of compressed sensing matrices which are then demonstrated to have associated sublinear-time sparse approximation algorithms. We then develop methods for constructing specialized matrices from this class which are sparse when multiplied with a discrete Fourier transform matrix. Ultimately, these considerations improve previous sampling requirements for deterministic sparse Fourier transform methods.

1. Introduction.

This paper considers methods for designing matrices which yield near-optimal nonlinear approximations to the Fourier transform of a given function, $f : [0, 2\pi] \to \mathbb{C}$. Suppose that f is a bandlimited function so that $\hat{f} \in \mathbb{C}^N$, where N is large. An optimal k-term trigonometric approximation to f is given by

$$f_k^{\mathrm{opt}}(x) = \sum_{j=1}^{k} \hat{f}\left(\omega_j\right) \mathbb{e}^{\mathbb{i}\omega_j x},$$

where $\omega_1, \ldots, \omega_N \in (-N/2, N/2] \cap \mathbb{Z}$ are ordered by the magnitudes of their Fourier coefficients so that

$$\left|\hat{f}(\omega_1)\right| \geq \left|\hat{f}(\omega_2)\right| \geq \cdots \geq \left|\hat{f}(\omega_N)\right|.$$

The optimal k-term approximation error is then

$$\left\|f - f_k^{\mathrm{opt}}\right\|_2 = \left\|\hat{f} - \hat{f}_k^{\mathrm{opt}}\right\|_2. \tag{1.1}$$

It has been demonstrated recently that any periodic function, $f : [0, 2\pi] \to \mathbb{C}$, can be accurately approximated via *sparse Fourier transform (SFT)* methods which run in $O(k^2 \log^4 N)$ time (see [26, 27] for details). When the function is sufficiently Fourier compressible (i.e., when a k which is much less than N yields a small approximation error in Equation (1.1) above), these methods can accurately approximate f much more quickly than traditional Fast Fourier Transform (FFT) methods which run in $O(N \log N)$ time. Furthermore, these SFT methods require only $O(k^2 \log^4 N)$ function evaluations as opposed to the N function evaluations required by a standard FFT method.

Although the theoretical guarantees of SFT algorithms appear promising, current algorithmic formulations suffer from several practical shortfalls. Principally, the algorithms currently utilize number theoretic sampling sets which are constructed in a suboptimal fashion. In this paper we address this deficiency by developing computational methods for constructing number theoretic matrices of the type required by these SFT methods which are nearly optimal in size. In the process, we demonstrate that this specific problem is a more constrained instance of a much more general matrix design problem with connections to compressed sensing matrix constructions [15, 9, 10, 8, 3], discrete uncertainty principles [18], nonadaptive group testing procedures [19, 22], and codebook design problems [42, 16, 7] in signal processing.

[*] Mathematics Department, Kansas State University, Manhattan, KS 66506, jamespb@ksu.edu, research supported in part by a McNair Scholarship and an I-Center Undergraduate Scholarship from Kansas State University.

[†] Contact Author. Mathematics Department, Duke University, Durham, NC 27708, markiwen@math.duke.edu, research supported in part by ONR N00014-07-1-0625 and NSF DMS-0847388.

[‡] Mathematics Department, Kansas State University, Manhattan, KS 66506, cvs@math.ksu.edu, research supported in part by NSA Young Investigators Grant H98230-10-1-0155.

1.1. General Problem Formulation: Compressed Sensing in the Fourier Setting.

Over the past several years, a stream of work in compressed sensing has provided a general theoretical framework for approximating general functions in terms of their optimal k-term approximation errors (see [20] and references therein). Indeed, the SFT design problem we are considering herein also naturally falls into this setting. Consider the following discretized version of the sparse Fourier approximation problem above: Let $\vec{f} \in \mathbb{C}^N$ be a vector of N equally spaced evaluations of f on $[0, 2\pi]$, and define $\mathcal{F}$ to be the $N \times N$ *Discrete Fourier Transform (DFT)* matrix defined by $\mathcal{F}_{i,j} = \frac{\mathbb{e}^{\frac{-2\pi \mathbb{i} \cdot i \cdot j}{N}}}{\sqrt{N}}$. Note that $\mathcal{F}\vec{f}$ will be compressible (i.e., sparse). Compressed sensing methods allow us to construct an $m \times N$ matrix, $\mathcal{M}$, with m minimized as much as possible subject to the constraint that an associated approximation algorithm, $\Delta_{\mathcal{M}} : \mathbb{C}^m \to \mathbb{C}^N$, can still accurately approximate any given $\hat{f} = \mathcal{F}\vec{f}$ (and, therefore, f itself). More exactly, compressed sensing methods allow us to minimize m, the number of rows in $\mathcal{M}$, as a function of k and N such that

$$\left\| \Delta_{\mathcal{M}}\left(\mathcal{M}\hat{f}\right) - \hat{f} \right\|_p \leq C_{p,q} \cdot k^{\frac{1}{p}-\frac{1}{q}} \left\| \hat{f} - \hat{f}_k^{\text{opt}} \right\|_q \tag{1.2}$$

holds for all $\hat{f} \in \mathbb{C}^N$ in various fixed l^p,l^q norms, $1 \leq q \leq p \leq 2$, for an absolute constant $C_{p,q} \in \mathbb{R}$ (e.g., see [13, 20]). Note that this implies that $\hat{f}$ will be recovered exactly if it contains only k nonzero Fourier coefficients. Similarly, it will be accurately approximated by $\Delta_{\mathcal{M}}\left(\mathcal{M}\hat{f}\right)$ any time it is well represented by its largest k Fourier modes.

In this paper we will focus on constructing $m \times N$ compressed sensing matrices, $\mathcal{M}$, for the Fourier recovery problem which meet the following four design requirements:

1. **Small Sampling Requirements:** $\mathcal{M}\mathcal{F}$ should be highly column-sparse (i.e., the number of columns of $\mathcal{M}\mathcal{F}$ which contain nonzero entries should be significantly smaller than N). Note that whenever $\mathcal{M}\mathcal{F}$ has this property we can compute $\mathcal{M}\hat{f}$ by reading only a small fraction of the entries in $\vec{f}$. Once the number of required function samples/evaluations is on the order of N, a simple fast Fourier transform based approach will be difficult to beat computationally.
2. **Accurate Approximation Algorithms:** The matrix $\mathcal{M}$ needs to have an associated approximation algorithm, $\Delta_{\mathcal{M}}$, which allows accurate recovery. More specifically, we will require an instance optimal error guarantee along the lines of Equation (1.2).
3. **Efficient Approximation Algorithms:** The matrix $\mathcal{M}$ needs to have an associated approximation algorithm, $\Delta_{\mathcal{M}}$, which is computationally efficient. In particular, the algorithm should be at least polynomial time in N (preferably, $o(N \log N)$-time since N is presumed to be large and we have the goal in mind of competing with an FFT).
4. **Guaranteed Uniformity:** Given only $k, N \in \mathbb{Z}^+$ and $p, q \in [1, 2]$, one fixed matrix $\mathcal{M}$ together with a fixed approximation algorithm $\Delta_{\mathcal{M}}$ should be guaranteed to satisfy the three proceeding properties uniformly for all vectors $\hat{f} \in \mathbb{C}^N$.

The remainder of this paper is organized as follows: We begin with a brief survey of recent sparse Fourier approximation techniques related to compressed sensing in Section 2. In Section 3 we introduce matrices of a special class which are useful for fast sparse Fourier approximation and investigate their properties. Most importantly, we demonstrate that any matrix from this class can be used in combination with an associated fast approximation algorithm in order to produce a sublinear-time (in N) compressed sensing method. Next, in Section 4, we present a deterministic construction of these matrices that specifically supports sublinear-time Fourier approximation. In Section 5 this matrix construction method is cast as an optimal design problem whose objective is to minimize Fourier sampling requirements. Furthermore, lemmas are proven which allow the optimal design problem to be subsequently formulated as an integer linear program in Section 6. Finally, in Section 7, we empirically investigate the sizes of the optimized deterministic matrices presented herein.

2. Compressed Sensing and The Restricted Isometry Property. Over the past few years, compressed sensing has focused primarily on utilizing matrices, $\mathcal{M}$, which satisfy the *Restricted Isometry Principle (RIP)* [9] in combination with l^1-minimization based approximation methods [9, 10, 8]. In fact, RIP matrices appear to be the critical partner in the RIP matrix/l^1-minimization pair since RIP matrices can also be used for compressed sensing with numerous other approximation algorithms besides l^1-minimization (e.g., Regularized Orthogonal Matching Pursuit [33, 34], CoSaMP [32], Iterative Hard Thresholding [5], etc.). Hence, we will consider RIP matrices in isolation.

Definition 1. *Let $p \in [1, \infty)$, $N, k \in \mathbb{N}$, and $\epsilon \in (0, 1)$. An $m \times N$ matrix $\mathcal{M}$ with complex entries has the Restricted Isometry Property, RIP$_p$(k,ϵ), if*

$$(1 - \epsilon)\,\|x\|_p^p \leq \|\mathcal{M}x\|_p^p \leq (1 + \epsilon)\,\|x\|_p^p$$

for all $x \in \mathbb{C}^N$ containing at most k nonzero coordinates.

It has been demonstrated that Fourier RIP$_2$(k,ϵ) matrices of size $O\left(k\frac{\log^4 N}{\epsilon^2}\right) \times N$ exist [38, 39]. More specifically, an $m \times N$ submatrix of the $N \times N$ *Inverse DFT (IDFT)* matrix, $\mathcal{F}^{-1}$, formed by randomly selecting m rows of $\mathcal{F}^{-1}$ will satisfy the RIP$_2$(k,ϵ) with high probability whenever m is $\Omega\left(k \log^2 N \frac{\log^2 k}{\epsilon^2}\right)$ [37]. Such a matrix will clearly satisfy our small sampling requirement since any $m \times N$ submatrix of the $N \times N$ IDFT matrix will generate a vector containing exactly m ones after being multiplied against the $N \times N$ DFT matrix. Furthermore, l^1-minimization will yield accurate approximation of Fourier compressible signals when utilized in conjunction with an IDFT submatrix that has the RIP$_2$. However, these random Fourier RIP$_2$ constructions have two deficiencies: First, all existing approximation algorithms, $\Delta_{\mathcal{M}}$, associated with Fourier RIP$_2$(k,ϵ) matrices, $\mathcal{M}$, run in $\Omega\left(N \log N\right)$ time. Thus, they cannot generally compete with an FFT computationally. Second, randomly generated Fourier submatrices are only guaranteed to have the RIP$_2$ with high probability, and there is no tractable means of verifying that a given matrix has the RIP$_2$. In order to verify Definition 1 for a given $m \times N$ matrix one generally has to compute the condition numbers of all $\binom{N}{k}$ of its $m \times k$ submatrices.

Several deterministic RIP$_2$(k,ϵ) matrix constructions exist which simultaneously address the guaranteed uniformity requirement while also guaranteeing small Fourier sampling needs [28, 6]. However, they all utilize the notion of *coherence* [15] which is discussed in Section 2.2. Hence, we will postpone a more detailed discussion of these methods until later. For now, we simply note that no existing deterministic RIP$_2$(k,ϵ) matrix constructions currently achieve a number of rows (or sampling requirements), m, that are $o\left(k^2 \operatorname{polylog}(N)\right)$ for all $k = o\left(\sqrt{N}\right)$ as N grows large. In contrast, RIP matrix constructions related to highly unbalanced expander graphs can currently break this "quadratic-in-k bottleneck".

2.1. Unbalanced Expander Graphs. Recently it has been demonstrated that the rescaled adjacency matrix of any unbalanced expander graph will be a RIP$_1$ matrix [3, 4].

Definition 2. *Let $N, k, d \in \mathbb{N}$, and $\epsilon \in (0, 1)$. A simple bipartite graph $G = (A, B, E)$ with $|A| \geq |B|$ and left degree at least d is a (k, d, ϵ)-unbalanced expander if, for any $X \subset A$ with $|X| \leq k$, the set of neighbors, $|N(X)|$, of X has size $|N(X)| \geq (1 - \epsilon)d|X|$.*

Theorem 1. ***(See [3, 4]).*** *Consider an $m \times N$ matrix $\mathcal{M}$ that is the adjacency matrix of a left-regular (k, d, ϵ)-unbalanced expander, where $1/\epsilon$ and d are both smaller than N. Then, there exists an absolute constant $C > 1$ such that the rescaled matrix, $\mathcal{M} \,/\, d^{1/p}$, satisfies the RIP$_p$($k$,$C\epsilon$) for all $1 \leq p \leq 1 + 1/\log N$.*

Note that the RIP_1 property for unbalanced expanders is with respect to the l^1 norm, not the l^2 norm. Nevertheless, matrices with the RIP_1 property also have associated approximation algorithms that can produce accurate sparse approximations along the lines of Equation (1.2) with $p = q = 1$. Examples include l^1-minimization [3, 4] and Matching Pursuit [25]. Perhaps most impressive among the approximation algorithms associated with unbalanced expander graphs are those which appear to run in $o(N \log N)$-time (see the appendix of [4]). Considering these results with respect to the four design requirements from Section 1.1, we can see that expander based RIP methods are poised to satisfy both the second and third requirements. Furthermore, by combining Theorem 1 with recent explicit constructions of unbalanced expander graphs [23], we can obtain an explicit RIP_1 matrix construction of near-optimal dimensions (which, among other things, shows that RIP_1 matrices may also satisfy our fourth Section 1.1 design requirement regarding guaranteed uniformity). We have the following theorem:

Theorem 2. *Let $\epsilon \in (0, 1)$, $p \in [1, 1 + 1/\log N]$, and $N, k \in \mathbb{N}$ be such that N is greater than both $1/\epsilon$ and k. Next, choose any constant parameter $\alpha \in \mathbb{R}^+$. Then, there exists a constant $c \in \mathbb{R}^+$ such that a*

$$O\left(k^{1+\alpha} (\log N \log k/\epsilon)^{2+2/\alpha}\right) \times N$$

matrix guaranteed to have the $\text{RIP}_p(k,\epsilon)$ can be constructed in $O\left(N \cdot (\log N/\epsilon)^{c(1+1/\alpha)}\right)$-time.

Proof: Consider Theorem 1.3 in [23] in combination with Theorem 1 above. □

Theorem 2 demonstrates the existence of deterministically constructible RIP_1 matrices with a number of rows, m, that scales like $O\left(k^{1+\alpha}\,\text{polylog}(N)\right)$ for all $k < N$ and fixed $\epsilon, \alpha \in (0, 1)$.[1] Furthermore, the run time complexity of the RIP_1 construction algorithm is modest (i.e., $O\left(N^2\right)$-time). Although a highly attractive result, there is no guarantee that Guruswami et al.'s unbalanced expander graphs will generally have adjacency matrices, $\mathcal{M}$, which are highly column-sparse after multiplication against a DFT matrix (see design requirement number 1 in Section 1.1).[2] Hence, it is unclear whether expander graph based RIP_1 results can be utilized to make progress on our compressed sensing matrix design problem in the Fourier setting. Nevertheless, this challenging avenue of research appears potentially promising, if not intractably difficult.

2.2. Incoherent Matrices.

As previously mentioned, all deterministic $\text{RIP}_2(k,\epsilon)$ matrix constructions (e.g., see [17, 28, 37, 6] and references therein) currently utilize the notion of coherence [15].

Definition 3. *Let $\mu \in [0, 1]$. An $m \times N$ matrix, $\mathcal{M}$, with complex entries is called μ-coherent if both of the following properties hold:*

1. *Every column of $\mathcal{M}$, denoted $\mathcal{M}_{\cdot,j} \in \mathbb{C}^m$ for $0 \leq j \leq N - 1$, is normalized so that $\|\mathcal{M}_{\cdot,j}\|_2 = 1$.*
2. *For all $j, l \in [0, N)$ with $j \neq l$, the associated columns $\mathcal{M}_{\cdot,j}, \mathcal{M}_{\cdot,l} \in \mathbb{C}^m$ have $\left|\mathcal{M}_{\cdot,j} \cdot \mathcal{M}_{\cdot,l}\right| \leq \mu$.*

Theorem 3. ***(See [37]).*** *Suppose that an $m \times N$ matrix, $\mathcal{M} \in \mathbb{C}^{m\times N}$, is μ-coherent. Then, $\mathcal{M}$ will also have the $\text{RIP}_2(k,(k-1)\mu)$.*

[1] Only values of α that scale like $\Omega\left(\log\log N/\log N\right)$ can result in RIP_1 matrices with fewer rows than columns.

[2] In fact, $\mathcal{M}$ multiplied against a DFT matrix need not be exactly sparse. By appealing to ideas from [24], one can see that it is enough to have a relatively small perturbation of $\mathcal{M}$ be column-sparse after multiplication against a DFT matrix.

Matrices with small coherence are of interest in numerous coding theoretic settings. Note that the column vectors of a real valued matrix with small coherence, μ, collectively form a spherical code. More generally, the columns of an incoherent complex valued matrix can be used to form codebooks for various channel coding applications in signal processing [31, 41]. These applications have helped to motivate a considerable amount of work with incoherent codes (i.e., incoherent matrices) over the past several decades. As a result, a plethora of μ-coherent matrix constructions exist (e.g., see [42, 16, 7, 6], and references therein).

As we begin to demonstrate in the next section, matrices with low coherence can satisfy all four Fourier design requirements listed in Section 1.1. However, there are trade-offs. Most notably, the Welch bound [42] implies that any μ-coherent $m \times N$ matrix, $\mathcal{M} \in \mathbb{C}^{m \times N}$, must have a number of rows

$$m \geq \frac{N}{(N-1)\mu^2 + 1}.$$

As a consequence, arguments along the lines of Theorem 3 can only use μ-coherent matrices to produce $\text{RIP}_2(k,\epsilon)$ matrices having $m = \Omega\left(k^2/\epsilon^2\right)$ rows. In contrast, $O\left(k\frac{\log^4 N}{\epsilon^2}\right) \times N$ Fourier $\text{RIP}_2(k,\epsilon)$ matrices are known to exist (see above). Hence, although μ-coherent matrices do allow one to obtain small Fourier sampling requirements, these sampling requirements all currently scale *quadratically* with k instead of *linearly*.[3]

Setting aside the quadratic scaling of m with k, we can see that several existing deterministic $\text{RIP}_2(k,\epsilon)$ matrix constructions based on coherence arguments (e.g., [28, 6]) immediately satisfy all but one of the Fourier design requirements listed in Section 1.1. First, these constructions lead to Fourier sampling requirements which, although generally quadratic in the sparsity parameter k, are nonetheless $o(N)$. Second, these matrices can be used in conjunction with accurate approximation algorithms (e.g., l^1-minimization) since they will have the RIP_2. Third, the deterministic nature of these RIP_2 matrices guarantees uniform approximation results for all possible periodic functions. The only unsatisfied design requirement pertains to the computational efficiency of the approximation algorithms (see requirement 3 in Section 1.1). As mentioned previously, all existing approximation algorithms associated with Fourier $\text{RIP}_2(k,\epsilon)$ matrices run in $\Omega\left(N \log N\right)$ time. In the next section we will present a general class of incoherent matrices which have fast approximation algorithms associated with them. As a result, we will develop a general framework for constructing fast sparse Fourier algorithms which are capable of approximating compressible signals more quickly than standard FFT algorithms.

3. A Special Class of Incoherent Matrices.

In this section, we will consider binary incoherent matrices, $\mathcal{M} \in \{0,1\}^{m \times N}$, as a special subclass of incoherent matrices. As we shall see, binary incoherent matrices can be used to construct RIP_2 matrices (e.g., via Theorem 3), unbalanced expander graphs (and, therefore, $\text{RIP}_{p\approx 1}$ matrices via Theorem 1), and nonadaptive group testing matrices [19]. In addition, we prove that any binary incoherent matrix can be modified to have an associated accurate approximation algorithm, $\Delta_{\mathcal{M}} : \mathbb{C}^m \to \mathbb{C}^N$, with sublinear $o(N)$ run time complexity. This result generalizes the fast sparse Fourier transforms previously developed in [27] to the standard compressed sensing setup while simultaneously providing a framework for the subsequent development of similar Fourier results. We will begin this process by formally defining (K, α)-coherent matrices and then noting some accompanying bounds.

[3] It is worth noting that Bourgain et al. recently used methods from additive combinatorics in combination with modified coherence arguments to construct explicit $m \times N$ matrices, with $m = O(k^{2-\epsilon'})$, which have the Fourier $\text{RIP}_2(k,m^{-\epsilon'})$ whenever $k = \Omega(N^{1/2-\epsilon'})$ [6]. Here $\epsilon' > 0$ is some constant real number. Hence, it is possible to break the previously mentioned "quadratic bottleneck" for $\text{RIP}_2(k,\epsilon)$ matrices when k is sufficiently large.

Definition 4. *Let $K, \alpha \in [1, m] \cap \mathbb{N}$. An $m \times N$ binary matrix, $\mathcal{M} \in \{0,1\}^{m \times N}$, is called (K, α)-coherent if both of the following properties hold:*

1. *Every column of $\mathcal{M}$ contains at least K nonzero entries.*
2. *For all $j, l \in [0, N)$ with $j \neq l$, the associated columns, $\mathcal{M}_{\cdot,j}$ and $\mathcal{M}_{\cdot,l} \in \{0,1\}^m$, have $\mathcal{M}_{\cdot,j} \cdot \mathcal{M}_{\cdot,l} \leq \alpha$.*

Several deterministic constructions for (K, α)-coherent matrices have been implicitly developed as part of RIP_2 matrix constructions (e.g., see [17, 28]). It is not difficult to see that any (K, α)-coherent matrix will be $\frac{\alpha}{K}$-coherent after having its columns normalized. Hence, the Welch bound also applies to (K, α)-coherent matrices. Below we will both develop tighter lower row bounds, and provide a preliminary demonstration of the existence of fast $o(N)$-time compressed sensing algorithms related to incoherent matrices. This will be done by demonstrating the relationship between (K, α)-coherent matrices and group testing matrices.

3.1. Group Testing: Lower Bounds and Fast Recovery.

Group testing generally involves the creation of testing procedures which are designed to identify a small number of interesting items hidden within a much larger set of uninteresting items [19, 22]. Suppose we are given a collection of N items, each of which is either interesting or uninteresting. The status of each item in the set can then be represented by a boolean vector $\vec{x} \in \{0,1\}^N$. Interesting items are denoted with a 1 in the vector, while uninteresting items are marked with a 0. Because most items are uninteresting, $\vec{x}$ will contain at most a small number, $d < N$, of ones. Our goal is to correctly identify the nonzero entries of $\vec{x}$, thereby recovering $\vec{x}$ itself.

Consider the following example. Suppose that $\vec{x}$ corresponds to a list of professional athletes, at most d of which are secretly using a new performance enhancing drug. Furthermore, imagine that the only test for the drug is an expensive and time consuming blood test. The trivial solution would be to collect blood samples from all N athletes and then test each blood sample individually for the presence of the drug. However, this is unnecessarily expensive when the test is accurate and the number of drug users is small. A cheaper solution involves pooling portions of each player's blood into a small number of well-chosen testing pools. Each of these testing pools can then be tested once, and the results used to identify the offenders.

A pooling-based testing procedure as described above can be modeled mathematically as a boolean matrix $\mathcal{M} \in \{0,1\}^{m \times N}$. Each row of $\mathcal{M}$ corresponds to a subset of the N athletes' whose blood will be pooled, mixed, and then tested once for the presence of the drug. Hence, the goal of our nonadaptive group testing can be formulated at follows: Design a matrix, $\mathcal{M} \in \{0,1\}^{m \times N}$, with as few rows as possible so that any boolean vector, $\vec{x} \in \{0,1\}^N$, containing at most d nonzero entries can be recovered exactly from the result of the pooled tests, $\mathcal{M}\vec{x} \in \{0,1\}^m$. Here all arithmetic is boolean, with the boolean *OR* operator replacing summation and the boolean *AND* operator replacing multiplication. One well studied solution to this nonadaptive group testing problem is to let $\mathcal{M}$ be a d-disjunct matrix.

Definition 5. *An $m \times N$ binary matrix, $\mathcal{M} \in \{0,1\}^{m \times N}$, is called d-disjunct if for any subset of $d+1$ columns of $\mathcal{M}$, $C = \{c_1, c_2, \ldots, c_{d+1}\} \subset [1, N] \cap \mathbb{N}$, there exists a subset of $d+1$ rows of $\mathcal{M}$, $R = \{j_1, j_2, \ldots, j_{d+1}\} \subset [1, m] \cap \mathbb{N}$, such that the submatrix*

$$\begin{pmatrix} \mathcal{M}_{j_1,c_1} & \mathcal{M}_{j_1,c_2} & \ldots & \mathcal{M}_{j_1,c_{d+1}} \\ \mathcal{M}_{j_2,c_1} & \mathcal{M}_{j_2,c_2} & \ldots & \mathcal{M}_{j_2,c_{d+1}} \\ & & \vdots & \\ \mathcal{M}_{j_{d+1},c_1} & \mathcal{M}_{j_{d+1},c_2} & \ldots & \mathcal{M}_{j_{d+1},c_{d+1}} \end{pmatrix}.$$

is the $(d+1) \times (d+1)$ identity matrix.[4]

[4] This is not the standard statement of the definition. Traditionally, a boolean matrix $\mathcal{M}$ is said to be d-disjunct if

Nonadaptive group testing is closely related to the recovery of "exactly sparse" vectors $\vec{x} \in \mathbb{R}^N$ containing exactly d nonzero entries. In fact, it is not difficult to modify standard group testing techniques to solve such problems. However, it is not generally possible to modify these approaches in order to obtain methods capable of achieving the type of approximation guarantees we are interested in here (i.e., see Equation (1.2)). However, fast $o(N)$-time approximation algorithms based on d-disjunct matrices with weaker approximation guarantees have been developed [14]. Hence, if we can relate (K, α)-coherent matrices to d-disjunct matrices, we will informally settle the design requirement regarding the existence of fast approximation algorithms (see the third design requirement in Section 1.1).

Lemma 1. *An $m \times N$ (K, α)-coherent matrix, $\mathcal{M}$, will also be $\lfloor (K-1)/\alpha \rfloor$-disjunct.*

Proof: Choose any subset of $\lfloor (K-1)/\alpha \rfloor + 1$ columns from $\mathcal{M}$, $C = \{c_1, c_2, \dots, c_{\lfloor (K-1)/\alpha \rfloor + 1}\} \subset [1, N] \cap \mathbb{N}$. Consider the column $\mathcal{M}_{\cdot, c_1} \in \{0,1\}^m$. Because $\mathcal{M}$ is a binary (K, α)-coherent matrix, we know that there can be at most α rows, j, for which $\mathcal{M}_{j, c_1} = \mathcal{M}_{j, c_2} = 1$. Hence, there are at most $\alpha \lfloor (K-1)/\alpha \rfloor \le K - 1$ total rows in which $\mathcal{M}_{\cdot, c_1}$ will share a 1 with any of the other columns listed in C. Since $\mathcal{M}_{\cdot, c_1}$ contains at least K ones, there exists a row, $j_1 \in [1, m] \cap \mathbb{N}$, containing a 1 in column c_1 and zeroes in all of $C - \{c_1\}$. Repeating this argument with $c_2, \dots, c_{\lfloor (K-1)/\alpha \rfloor + 1}$ replacing c_1 above proves the lemma. □

Any $m \times N$ d-disjunct matrix must have $m = \Omega\left(\min\{d^2 \log_d N, N\}\right)$ [12]. Furthermore, near-optimal explicit d-disjunct measurement matrix constructions of size $O(d^2 \log N) \times N$ exist [36]. Of more interest here, however, is that the lower bound for d-disjunct matrices together with Lemma 1 provides a lower bound for (K, α)-coherent matrices. More specifically, we can see that any $m \times N$ (K, α)-coherent matrix must have $m = \Omega\left(\min\left\{\frac{K^2}{\alpha^2} \log_{K/\alpha} N, N\right\}\right)$.

3.2. Properties of Binary Incoherent Matrices. We begin this section by presenting a simple sublinear-time recovery algorithm, $\Delta_{\mathcal{M}}$, which is guaranteed to satisfy an approximation guarantee along the lines of Equation 1.2 for all (K, α)-coherent matrices, $\mathcal{M}$, and vectors $\vec{x} \in \mathbb{C}^N$. In what follows we will generalize recovery methods and matrix constructions from [27]. We will give self-contained arguments whenever possible, although it will be necessary on occasion to state generalized results from [27] whose proofs we omit.

Before we can continue we must establish some notation. For any given $\vec{x} \in \mathbb{C}^N$ and subset $S \subseteq [0, N) \cap \mathbb{N}$, we will let $\vec{x}_S \in \mathbb{C}^N$ be equal to $\vec{x}$ on the indexes in S and be zero elsewhere. Thus,

$$(\vec{x}_S)_i = \begin{cases} x_i & \text{if } i \in S \\ 0 & \text{otherwise} \end{cases}.$$

Furthermore, for a given integer $k < N$, we will let $S_k^{\text{opt}} \subset [0, N) \cap \mathbb{N}$ be the first k element subset of $[0, N) \cap \mathbb{N}$ in lexicographical order with the property that $|x_s| \ge |x_t|$ for all $s \in S_k^{\text{opt}}$ and $t \in [0, N) \cap \mathbb{N} - S_k^{\text{opt}}$. Thus, S_k^{opt} contains the indexes of k of the largest magnitude entries in $\vec{x}$. Finally, we will define $\vec{x}_k^{\text{opt}}$ to be $\vec{x}_{S_k^{\text{opt}}}$, an optimal k-term approximation to $\vec{x}$. The following lemma provides the basis of our sublinear-time recovery method.

Lemma 2. *Suppose $\mathcal{M}$ is a (K, α)-coherent matrix. Let $n \in [0, N) \cap \mathbb{N}$, $k \in \left[1, \frac{K}{\alpha}\right] \cap \mathbb{N}$, $\epsilon \in (0, 1]$, $c \in [2, \infty) \cap \mathbb{N}$, and $\vec{x} \in \mathbb{C}^N$. Finally, let $\mathcal{M}(K, n)$ be the $K \times N$ submatrix of $\mathcal{M}$ created by selecting the first K rows of $\mathcal{M}$ with nonzero entries in the n^{th} column. Then, if $K >$*

the boolean *OR* of any d of its columns does not contain any other column [19, 22]. However, these two definitions are essentially equivalent. The d-disjunct condition is also equivalent to the $(d+1)$-strongly selective condition utilized by compressed sensing algorithms based on group testing matrices [14].

$c\cdot(k\alpha/\epsilon)$, $(\mathcal{M}(K,n)\cdot\vec{x})_j$ will be contained in the interval $\left(x_n - \frac{\epsilon\left\|\vec{x}-\vec{x}^{\rm opt}_{(k/\epsilon)}\right\|_1}{k},\ x_n + \frac{\epsilon\left\|\vec{x}-\vec{x}^{\rm opt}_{(k/\epsilon)}\right\|_1}{k}\right)$ for more than $\frac{c-2}{c}\cdot K$ values of $j \in [0,K)\cap\mathbb{N}$.
Proof: See Appendix A. □

Consider Lemma 2 with $c = 4$ for a given $m\times N$ (K,α)-coherent matrix $\mathcal{M}$, $k \in [1, K/\alpha]\cap\mathbb{N}$, and $\epsilon \in (0,1]$. Let $\vec{x} \in \mathbb{C}^N$ and suppose that

$$x_n > 2\delta = 2\left(\frac{\epsilon\left\|\vec{x}-\vec{x}^{\rm opt}_{(k/\epsilon)}\right\|_1}{k}\right) \tag{3.1}$$

for some $n \in [0,N)\cap\mathbb{N}$. We will begin describing our sublinear-time recovery method by demonstrating a means of identifying n using only the measurements $\mathcal{M}\vec{x} \in \mathbb{C}^m$ together with some additional linear measurements based on a modification of our binary incoherent matrix $\mathcal{M}$. This technique, first utilized in [14], will ultimately allow us to develop the sublinear-time approximation schemes we seek. However, we require several definitions before we can continue with our demonstration.

Let $\mathcal{A} \in \mathbb{R}^{m\times N}$ and $\mathcal{C} \in \mathbb{R}^{\tilde{m}\times N}$ be real matrices. Then, their row tensor product, $\mathcal{A}\circledast\mathcal{C}$, is defined to be the $(m\cdot\tilde{m})\times N$ matrix whose entries are given by

$$(\mathcal{A}\circledast\mathcal{C})_{i,j} = \mathcal{A}_{i \text{ mod m}, j}\cdot\mathcal{C}_{\frac{(i-(i \text{ mod m}))}{m}, j}.$$

We will use the row tensor product of $\mathcal{M}$ with the $\left(1+\lceil\log_2 N\rceil\right)\times N$ *bit test matrix* [14, 22] to help us identify n from Equation 3.1.[5] The $\left(1+\lceil\log_2 N\rceil\right)\times N$ bit test matrix, $\mathcal{B}_N$, is defined by

$$(\mathcal{B}_N)_{i,j} = \begin{cases} 1 & \text{if } i = 0 \\ (i-1)^{\rm th} \text{ bit in the binary expansion of } j & \text{if } i \in \left[1, \lceil\log_2 N\rceil\right] \end{cases} \tag{3.2}$$

for $0 \le i \le \lceil\log_2 N\rceil$ and $0 \le j < N$. For example, $\mathcal{B}_8$ has the form

$$\mathcal{B}_8 = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{pmatrix}.$$

We will now demonstrate that $(\mathcal{M}\circledast\mathcal{B}_N)\,\vec{x}$ contains enough information for us to identify any $n \in [0,N)\cap\mathbb{N}$ satisfying Equation 3.1.

Notice that $\mathcal{M}\circledast\mathcal{B}_N$ contains $\mathcal{M}$ as a submatrix. This is due to the first row of all ones in $\mathcal{B}_N$. Similarly, the second row of $\mathcal{B}_N$ ensures that $\mathcal{M}\circledast\mathcal{B}_N$ will contain another $m\times N$ submatrix which is identical to $\mathcal{M}$, except with all of its even columns zeroed out. We will refer to this $m\times N$ submatrix of $\mathcal{M}\circledast\mathcal{B}_N$ as $\mathcal{M}_{\rm odd}$. We can see that

$$\mathcal{M}_{\rm odd} \;=\; \mathcal{M}\circledast(\mathcal{B}_N)_1 \;=\; \left(\vec{0}\;\; \mathcal{M}_{\cdot,1}\;\; \vec{0}\;\; \mathcal{M}_{\cdot,3}\;\; \vec{0}\;\; \mathcal{M}_{\cdot,5}\;\; \vec{0}\;\; \ldots\right).$$

[5] We could also use the number theoretic $\mathcal{N}_{\lambda,s_1}$ matrices defined in Section 5 of [27] here in place of the bit test matrix. More generally, any 1-disjunct matrix with an associated fast decoding algorithm could replace the bit test matrix throughout this section. Note that a fast $O(t)$-time binary tree decoder can be built for any $t\times N$ 1-disjunct matrix anytime one has access to $\Omega(N)$ memory.

Furthermore, we define

$$\mathcal{M}_{\text{even}} \ := \ \mathcal{M} \ - \ \mathcal{M} \circledast (\mathcal{B}_N)_1 \ = \ \mathcal{M} \ - \ \mathcal{M}_{\text{odd}}.$$

Clearly, if we are given $(\mathcal{M} \circledast \mathcal{B}_N)\,\vec{x}$, we will also have $\mathcal{M}\vec{x}$, $\mathcal{M}_{\text{odd}}\vec{x}$, and $\mathcal{M}_{\text{even}}\vec{x} \in \mathbb{C}^m$. We can use this information to determine whether n from Equation 3.1 is even or odd as follows.

Lemma 2 with $c = 4$ guarantees that more than $K/2$ distinct elements of $\mathcal{M}\vec{x} \in \mathbb{C}^m$ will be of the form

$$(\mathcal{M}\vec{x})_j \ = \ x_n \cdot \mathcal{M}_{j,n} + \gamma \tag{3.3}$$

for some $j \in [1, m]$ and $\gamma \in \mathbb{C}$ with $\left|\gamma\right| \leq \delta$ (see Equation 3.1). Suppose n is odd. Then, for each j satisfying Equation 3.3, we will have

$$\left|(\mathcal{M}_{\text{even}}\vec{x})_j\right| \leq \delta < |x_n| - \left|\gamma\right| \leq \left|(\mathcal{M}_{\text{odd}}\vec{x})_j\right|.$$

Similarly, if n is even, then for each such j we will have

$$\left|(\mathcal{M}_{\text{odd}}\vec{x})_j\right| \leq \delta < |x_n| - \left|\gamma\right| \leq \left|(\mathcal{M}_{\text{even}}\vec{x})_j\right|.$$

Therefore, we can correctly determine n mod 2 by comparing $\left|(\mathcal{M}_{\text{odd}}\vec{x})_j\right|$ with $\left|(\mathcal{M}_{\text{even}}\vec{x})_j\right|$ whenever both Equations 3.1 and 3.3 hold. Of course, there is nothing particularly special about the lowest order bit of the binary representation of n. More generally, we can correctly determine the i^{th} bit of $n \in [0, N) \cap \mathbb{N}$ by comparing $\left|(\mathcal{M} \circledast (\mathcal{B}_N)_{i+1}\,\vec{x})_j\right|$ with $\left|[(\mathcal{M} \ - \ \mathcal{M} \circledast (\mathcal{B}_N)_{i+1})\,\vec{x}]_j\right|$ whenever both Equations 3.1 and 3.3 hold.

We now know that we can correctly determine n whenever both Equations 3.1 and 3.3 hold by finding its binary representation one bit at a time. Furthermore, Lemma 2 with $c \geq 4$ guarantees that more than $K/2$ of the $j \in [1, m]$ will satisfy Equation 3.3 for any given n. Hence, we can correctly reconstruct every n for which Equation 3.1 holds more than $K/2$ times by attempting to decode n's binary representation for all $j \in [1, m]$. Furthermore, because $c \geq 4$, Lemma 2 guarantees that more than half of the K entries of $\mathcal{M}(K, n) \cdot \vec{x}$ can accurately estimate x_n once n has been found. This is enough to guarantee that the imaginary part of x_n will be accurately estimated by the median of the imaginary parts of all K entries of $\mathcal{M}(K, n) \cdot \vec{x}$. Of course, the real part of x_n can also be estimated in a similar fashion. Hence, given both $\mathcal{M}$ and $\mathcal{M}\vec{x} \in \mathbb{C}^m$, Lemma 2 ensures that computing 2 medians of K elements of $\mathcal{M}(K, n) \cdot \vec{x}$ will allow us to accurately estimate x_n for any given $n \in [0, N) \cap \mathbb{N}$. Utilizing these methods we obtain Algorithm 1.

In light of the preceding discussion, we can see that Algorithm 1 will be guaranteed to identify all $n \in [0, N) \cap \mathbb{N}$ that satisfy Equation 3.1 more than $\frac{K}{2}$ times each. Lemma 2 can then be used to estimate x_n for each of these n values. The end result is that all relatively large entries in $\vec{x}$ will be identified and accurately estimated. By formalizing the discussion above, we obtain the following result, the proof of which is analogous to the proof of Theorem 7 in Section 5 of [27].

Theorem 3. *Suppose $\mathcal{M}$ is an $m \times N$ (K, α)-coherent matrix. Furthermore, let $\epsilon \in (0, 1]$, $k \in \left[1, K \cdot \frac{\epsilon}{4\alpha}\right) \cap \mathbb{N}$, and $\vec{x} \in \mathbb{C}^N$. Then, Algorithm 1 will output a $\vec{z}_S \in \mathbb{C}^N$ satisfying*

$$\left\|\vec{x} - \vec{z}_S\right\|_2 \ \leq \ \left\|\vec{x} - \vec{x}_k^{\,\text{opt}}\right\|_2 + \frac{22\epsilon \left\|\vec{x} - \vec{x}_{(k/\epsilon)}^{\,\text{opt}}\right\|_1}{\sqrt{k}}.$$

Algorithm 1 can be implemented to run in $O\left(m \log N\right)$ time.

Algorithm 1 Approximate $\vec{x}$

```
 1: Input: An $m \times N$ $(K, \alpha)$-coherent matrix, $\mathcal{M}$, and $(\mathcal{M} \circledast \mathcal{B}_N)\, \vec{x} \in \mathbb{C}^{m\lceil \log_2 N \rceil + m}$
 2: Output: $\vec{z}_S \approx \vec{x}_S$, an approximation to $\vec{x}_k^{\,\rm opt}$
 3: Initialize multiset $S \leftarrow \emptyset$, $\vec{z} \leftarrow \vec{0}_N$, $\vec{b} \leftarrow \vec{0}_{\lceil \log_2 N \rceil}$
                 Identify All $n \in [0, N) \cap \mathbb{N}$ that Satisfy Equation 3.1
 4: for $j$ from 1 to $m$ do
 5:     for $i$ from 0 to $\lceil \log_2 N \rceil - 1$ do
 6:         if $\left|\left(\mathcal{M} \circledast (\mathcal{B}_N)_{i+1}\, \vec{x}\right)_j\right| > \left|\left(\mathcal{M}\vec{x} - \mathcal{M} \circledast (\mathcal{B}_N)_{i+1}\, \vec{x}\right)_j\right|$ then
 7:             $b_i \leftarrow 1$
 8:         else
 9:             $b_i \leftarrow 0$
10:         end if
11:     end for
12:     $n \leftarrow \sum_{i=0}^{\lceil \log_2 N \rceil - 1} b_i 2^i$
13:     $S \leftarrow S \cup \{n\}$
14: end for
                 Estimate $\vec{x}_S \approx \vec{x}_k^{\,\rm opt}$ Using Lemma 2
15: for each $n$ value belonging to $S$ with multiplicity $> \frac{K}{2}$ do
16:     $\mathbb{Re}\,\{z_n\} \leftarrow$ median of multiset $\left\{\mathbb{Re}\,\{(\mathcal{M}(K,n) \cdot \vec{x})_h\} \;\middle|\; 1 \leq h \leq K\right\}$
17:     $\mathbb{Im}\,\{z_n\} \leftarrow$ median of multiset $\left\{\mathbb{Im}\,\{(\mathcal{M}(K,n) \cdot \vec{x})_h\} \;\middle|\; 1 \leq h \leq K\right\}$
18: end for
19: Sort nonzero $\vec{z}$ entries by magnitude so that $|z_{n_1}| \geq |z_{n_2}| \geq |z_{n_3}| \geq \dots$
20: $S \leftarrow \{n_1, n_2, \dots, n_{2k}\}$
21: Output: $\vec{z}_S$
```

The runtime of Algorithm 1 can be accounted for as follows: Lines 4 through 14 can be implemented to run in $O(m \log N)$ time since their execution time will be dominated by the time required to read each entry of $(\mathcal{M} \circledast \mathcal{B}_N)\, \vec{x} \in \mathbb{C}^{m\lceil \log_2 N \rceil + m}$. Counting the multiplicity of the $O(m)$ entries in S in line 15 can be done by sorting S in $O(m \log m)$ time, followed by one $O(m)$-time scan of the sorted data. Lines 16 and 17 will each be executed a total of $O(m/K)$ times apiece. Furthermore, lines 16 and 17 can each be executed in $O(K \log K)$ time assuming that each $\mathcal{M}(K,n)$ submatrix is known in advance.[6] Thus, the total runtime of lines 4 through 18 will also be $O(m \log N)$. Finally, line 19 requires that at most $O(m/K)$ items be sorted, which can likewise be accomplished in $O(m \log N)$ time. Therefore, the total runtime of Algorithm 1 will be $O(m \log N)$.

The following theorem summarizes several important properties of (K, α)-coherent matrices with respect to general sparse approximation problems.

Theorem 4. *Let $\mathcal{M}$ be an $m \times N$ (K, α)-coherent matrix. Then, all of the following statements will hold:*

1. *Let $\epsilon \in (0, 1]$, $k \in \left[1, K \cdot \frac{\epsilon}{4\alpha}\right) \cap \mathbb{N}$. There exists an approximation algorithm based on $\mathcal{M} \circledast \mathcal{B}_N$, $\Delta_{\mathcal{M}} : \mathbb{C}^{m\lceil \log_2 N \rceil + m} \rightarrow \mathbb{C}^N$, that is guaranteed to output a vector $\vec{z}_S \in \mathbb{C}^N$*

[6] If each $\mathcal{M}(K,n)$ submatrix is not computed once in advance this runtime will be $O(m \log m)$ instead of $O(K \log K)$.

satisfying

$$\left\|\vec{x} - \vec{z}_S\right\|_2 \;\leq\; \left\|\vec{x} - \vec{x}_k^{\,\mathrm{opt}}\right\|_2 + \frac{22\epsilon \left\|\vec{x} - \vec{x}^{\,\mathrm{opt}}_{(k/\epsilon)}\right\|_1}{\sqrt{k}} \tag{3.4}$$

for all $\vec{x} \in \mathbb{C}^N$. Most importantly, $\Delta_{\mathcal{M}}$ can be evaluated in $O\left(m \log N\right)$-time.

2. *Define the $m \times N$ matrix $\mathcal{W}$ by normalizing the columns of $\mathcal{M}$ so that $\mathcal{W}_{i,j} = \mathcal{M}_{i,j}/\sqrt{\|\mathcal{M}_{\cdot,j}\|_1}$. Then, the matrix $\mathcal{W}$ will be $\frac{\alpha}{K}$-coherent.*
3. *Furthermore, the $m \times N$ matrix $\mathcal{W}$ defined above will have the RIP$_2$(k,$(k-1)\alpha/K$).*[7]
4. *Define the $m \times N$ matrix $\mathcal{W}$ by $\mathcal{W}_{i,j} = \mathcal{M}_{i,j}/\left(\|\mathcal{M}_{\cdot,j}\|_1\right)^{\frac{1}{p}}$. Then, the matrix $\mathcal{W}$ will have the RIP$_p$(k,$C(k-1)\alpha/K$) for all $1 \leq p \leq 1 + \frac{1}{\log N}$, where C is an absolute constant larger than $1/2$.*
5. *$\mathcal{M}$ is $\lfloor (K-1)/\alpha \rfloor$-disjunct.*
6. *$\mathcal{M}$ has at least $m = \Omega\left(\min\left\{(K^2/\alpha^2)\log_{K/\alpha} N, N\right\}\right)$ rows.*

Proof: The proof of each part is as follows.

1. See Theorem 3.
2. The proof follows easily from the definitions.
3. The proof follows from part 2 together with Theorem 3.
4. Note that we can consider $\mathcal{M}$ to be the adjacency matrix of a bipartite graph, $G = (A, B, E)$, with $|A| = N$ and $|B| = m$. Each element of A will have degree at least K. Furthermore, for any $X \subset A$ with $|X| \leq k$ we can see that the set of neighbors of X will have

$$|N(X)| \geq \sum_{j=0}^{|X|-1} (K - j\alpha) \geq |X| \cdot K \cdot \left(1 - \frac{\alpha(|X|-1)}{2K}\right).$$

Hence, $\mathcal{M}$ is the adjacency matrix of a $(k, K, (k-1)\alpha/2K)$-unbalanced expander graph. The result now follows from the proof of Theorem 1 in [3].

Finally, the proof of parts 5 and 6 follow from Lemma 1 and the subsequent discussion in Section 3.1, respectively. □

To the best of our knowledge, the part 1 of Theorem 4 is the first deterministic sublinear-time Compressed Sensing (CS) result which both achieves an approximation guarantee along the lines of Equation (1.2), and has a runtime complexity which scales quadratically with sparsity, k.[8] The first deterministic sublinear-time CS results based on group testing matrices [14] obtained weaker min-max types of error guarantees over classes of compressible signals. Furthermore, the runtime of the resulting deterministic methods depended on the compressible signal class over which one was working. Fast deterministic CS algorithms based on expander graphs yield algorithms which require fewer measurements (i.e., a matrix $\mathcal{M}$ with fewer rows), but their runtime complexities scale like $k^{2+\Omega\left(\frac{\log\log N}{\log N}\right)} \cdot \mathrm{polylog}(N)$.[9] Randomized sublinear-time CS algorithms which obtain sparse approximation guarantees along the lines of Equation (1.2) for all signals simultaneously with high probability also exist (e.g.,

[7] It is worth noting that modified (K, α)-coherent matrices can also be used as Johnson-Lindenstrauss embeddings. See [30] together with [2] to learn more about the near equivalence of Johnson-Lindenstrauss embeddings and RIP$_2$ matrices.

[8] Part 6 of Theorem 4 implies that the number of rows in our matrices, m, must always scale quadratically in k.

[9] Consider the appendix of [4] together with Theorem 2 and the subsequent discussion.

see [21]). These methods match the $k^2 \cdot \text{polylog}(N)$ runtime complexity of part 1 of Theorem 4 while utilizing random matrices with fewer rows. However, none of the (random) matrices known to work in connection with [21, 4, 14] have small Fourier sampling requirements. Hence, the binary incoherent matrices considered herein are the only current option for obtaining sublinear-time methods with uniform recovery guarantees in the Fourier setting. Finally, we note that one may construct randomized sublinear-time CS methods based on binary incoherent matrices that both have $O\left(k \cdot \log^4 N\right)$ runtime complexities, and satisfy Equation 3.4 for each $\vec{x} \in \mathbb{C}^N$ with high probability (see the appendix in [1]).

Recall that explicit constructions of (K, α)-coherent matrices exist [17, 28]. It is worth noting that RIP$_2$ matrix constructions based on these (K, α)-coherent matrices are optimal in the sense that any RIP$_2$ matrix with binary entries must have a similar number of rows [11]. More interestingly, Theorem 4 formally demonstrates that (K, α)-coherent matrices satisfy all the Fourier design requirements in Section 1.1 other than the first one regarding small Fourier sampling requirements. In the sections below we will consider an optimized number theoretic construction for (K, α)-coherent matrices along the lines of the construction implicitly utilized in [28, 27]. As we shall demonstrate, these constructions have small Fourier sampling requirements. Hence, they will satisfy all four desired Fourier design requirements.

4. A (K, α)-Coherent Matrix Construction.

Let $\mathcal{F}_N$ denote the $N \times N$ unitary discrete Fourier transform matrix,

$$(\mathcal{F}_N)_{i,j} = \frac{\mathbb{e}^{\frac{-2\pi \mathbb{i} \cdot i \cdot j}{N}}}{\sqrt{N}}.$$

Recall that we want an $m \times N$ matrix $\mathcal{M}$ with the property that $\mathcal{M}\mathcal{F}_N$ contains nonzero values in as few columns as possible. In addition, we want $\mathcal{M}$ to be a binary (K, α)-coherent matrix so that we can utilize the sublinear-time approximation technique provided by Theorem 4. It appears to be difficult to achieve both of these goals simultaneously as stated. Hence, we will instead optimize a construction recently utilized in [27] which solves a trivial variant of this problem.

Let $\tilde{N}, N \in \mathbb{N}$ with $\tilde{N} > N$. We will say that an $m \times \tilde{N}$ matrix, $\widetilde{\mathcal{M}}$, is $(K, \alpha)_N$-coherent if the $m \times N$ submatrix of $\widetilde{\mathcal{M}}$ formed by its first N columns is (K, α)-coherent. In what follows we will consider ourselves to be working with $(K, \alpha)_N$-coherent matrices whose first N rows match a given $m \times N$ (K, α)-coherent matrix, $\mathcal{M}$, of interest. Note that this slight generalization will not meaningfully change anything previously discussed. For example, we may apply Theorem 4 to the submatrix formed by the first N columns of any given $(K, \alpha)_N$-coherent matrix, $\widetilde{\mathcal{M}}$, thereby effectively applying Theorem 4 to $\widetilde{\mathcal{M}}$ in the context of approximating vectors belonging to a fixed N-dimensional subspace of $\mathbb{C}^{\tilde{N}}$. The last $\tilde{N} - N$ columns of any $(K, \alpha)_N$-coherent matrix $\widetilde{\mathcal{M}}$ will be entirely ignored throughout this paper with one exception: We will hereafter consider it sufficient to guarantee that $\widetilde{\mathcal{M}}\mathcal{F}_{\tilde{N}}$ (as opposed to $\mathcal{M}\mathcal{F}_N$) contains nonzero values in as few columns as possible. This modification will not alter the sparse Fourier approximation guarantees (i.e., see Equation (1.1)) obtainable via Theorem 4 in any way when the functions being approximated are N-bandlimited. However, allowing $\tilde{N}$ to be greater than N will help us obtain small Fourier sampling requirements.

Let $\widetilde{\mathcal{M}}$ be an $m \times \tilde{N}$ $(K, \alpha)_N$-coherent matrix. It is useful to note that the column sparsity we desire in $\widetilde{\mathcal{M}}\mathcal{F}_{\tilde{N}}$ is closely related to the discrete uncertainty principles previously considered in [18].

Theorem 5. *(See [18]). Suppose $\vec{y} \in \mathbb{C}^{\tilde{N}}$ contains $\tilde{N}_t$ nonzero entries, while $\hat{y} = \vec{y}^{\,\mathrm{T}}\mathcal{F}_{\tilde{N}}$ contains $\tilde{N}_\omega$ nonzero entries. Then, $\tilde{N}_t\tilde{N}_\omega \geq \tilde{N}$. Furthermore, $\tilde{N}_t\tilde{N}_\omega = \tilde{N}$ holds if and only* *if $\vec{y}$ is a scalar multiple of a cyclic permutation of the binary picket fence sequence in $\mathbb{C}^{\tilde{N}}$ containing v equally-spaced nonzero elements*

$$(III^v)_u \ = \ \begin{cases} 1 & \textit{if}\ u \equiv 0\ \textit{mod}\ \frac{\tilde{N}}{v} \\ 0 & \text{otherwise} \end{cases},$$

where $v \in \mathbb{N}$ divides $\tilde{N}$.

We will build $m \times \tilde{N}$ $(K,\alpha)_N$-coherent matrices, $\mathcal{M}$, below whose rows are each a permuted binary picket fence sequence.[10] In this case Theorem 5 can be used to bound the number of columns of $\widetilde{\mathcal{M}\mathcal{F}_{\tilde{N}}}$ which contain nonzero entries. This, in turn, will bound the number of function samples required in order to approximate a given periodic bandlimited function.

We create an $m \times \tilde{N}$ $(K,\alpha)_N$-coherent matrix $\mathcal{M}$ as follows: Choose K pairwise relatively prime integers

$$s_1 < \cdots < s_K$$

and let $\tilde{N} = \prod_{j=1}^{K} s_j \ > \ N$. Next, we produce a picket fence row, $r_{j,h}$, for each $j \in [1,K] \cap \mathbb{N}$ and $h \in [0,s_j) \cap \mathbb{Z}$. Thus, the n^{th} entry of each row $r_{j,h}$ is given by

$$(r_{j,h})_n = \delta\left((n-h) \bmod s_j\right) = \begin{cases} 1 & \text{if } n \equiv h \bmod s_j \\ 0 & \text{otherwise} \end{cases}, \tag{4.1}$$

where $n \in [0,\tilde{N}) \cap \mathbb{Z}$. We then form $\mathcal{M}$ by setting

$$\mathcal{M} = \begin{pmatrix} r_{1,0} \\ r_{1,1} \\ \vdots \\ r_{1,s_1-1} \\ r_{2,0} \\ \vdots \\ r_{2,s_2-1} \\ \vdots \\ r_{K,s_K-1} \end{pmatrix}. \tag{4.2}$$

For an example measurement matrix see Figure 4.1.

Lemma 4. *An $m \times \tilde{N}$ matrix $\mathcal{M}$ as constructed in Equation (4.2) will be $(K,\lfloor \log_{s_1} N \rfloor)_N$-coherent with $m = \sum_{j=1}^{K} s_j$.*

Proof: Choose any two distinct integers, $l \neq n$, from $[0,N)$. Let $\mathcal{M}_{\cdot,l}$ and $\mathcal{M}_{\cdot,n}$ denote the l^{th} and n^{th} columns of $\mathcal{M}$, respectively. The inner product of these columns is

$$\mathcal{M}_{\cdot,l} \cdot \mathcal{M}_{\cdot,n} = \sum_{j=1}^{K} \delta\left((n-l) \bmod s_j\right).$$

The sum above is at most the maximum α for which $\prod_{j=1}^{\alpha} s_j \leq N$ by the Chinese Remainder Theorem. Furthermore, this value is itself bounded above by $\lfloor \log_{s_1} N \rfloor$. The equation for m immediately follows from the construction of $\mathcal{M}$ above. □

The following Lemma is a consequence of Theorem 5.

[10] We will refer to any sequence $III^v \in \mathbb{C}^{\tilde{N}}$ defined as in Theorem 5 as a "binary picket fence sequence".

$$\begin{array}{rl}
\mathbf{n \in [0, \tilde{N})} & \begin{array}{cccccccc} \mathbf{0} & \mathbf{1} & \mathbf{2} & \mathbf{3} & \mathbf{4} & \mathbf{5} & \mathbf{6} & \dots \end{array} \\
\begin{array}{r}
\mathbf{n \equiv 0\ (mod\ 2)} \\
\mathbf{n \equiv 1\ (mod\ 2)} \\
\mathbf{n \equiv 0\ (mod\ 3)} \\
\mathbf{n \equiv 1\ (mod\ 3)} \\
\mathbf{n \equiv 2\ (mod\ 3)} \\
\vdots \\
\mathbf{n \equiv 1\ (mod\ 5)} \\
\vdots
\end{array} &
\left(\begin{array}{cccccccc}
1 & 0 & 1 & 0 & 1 & 0 & 1 & \dots \\
0 & 1 & 0 & 1 & 0 & 1 & 0 & \dots \\
1 & 0 & 0 & 1 & 0 & 0 & 1 & \dots \\
0 & 1 & 0 & 0 & 1 & 0 & 0 & \dots \\
0 & 0 & 1 & 0 & 0 & 1 & 0 & \dots \\
 & & & \vdots & & & & \\
0 & 1 & 0 & 0 & 0 & 0 & 1 & \dots \\
 & & & \vdots & & & &
\end{array}\right)
\end{array}$$

FIG. 4.1. *Measurement Matrix, $\mathcal{M}$, Using $s_1 = 2$, $s_2 = 3$, $s_3 = 5, \dots$*

Lemma 5. *Let $\mathcal{M}$ be an $m \times \tilde{N}$ matrix as constructed in Equation (4.2). Then, $\mathcal{M}\mathcal{F}_{\tilde{N}}$ will contain nonzero entries in exactly $m - K + 1 = \left(\sum_{j=1}^{K} s_j\right) - K + 1$ columns.*

Proof: Fix $j \in [1, K] \cap \mathbb{N}$. Each picket fence row, $r_{j,h} \in \{0, 1\}^{\tilde{N}}$, contains $\tilde{N}/s_j$ ones. Thus, $r^{\mathrm{T}}_{j,h}\mathcal{F}_{\tilde{N}}$ contains s_j nonzero entries for all $h \in [0, s_j) \cap \mathbb{Z}$. Furthermore, $r^{\mathrm{T}}_{j,h}\mathcal{F}_{\tilde{N}}$ contains nonzero values *in the same entries for all $h \in [0, s_j) \cap \mathbb{Z}$* since all $r_{j,h}$ rows (with j fixed) are cyclic permutations of one another. Finally, let $l, j \in [1, K] \cap \mathbb{N}$ with $j \neq l$ and suppose that $r^{\mathrm{T}}_{j,h}\mathcal{F}_{\tilde{N}}$ and $r^{\mathrm{T}}_{l,g}\mathcal{F}_{\tilde{N}}$ both have nonzero values in the same entry. This can only happen if

$$h\frac{\tilde{N}}{s_j} = g\frac{\tilde{N}}{s_l}$$

for a pair of integers $0 \leq h < s_j$ and $0 \leq g < s_l$. However, since s_j and s_l are relatively prime, Euclid's lemma implies that this can only happen when $h = g = 0$. The result follows. □

We can now see that the matrix construction presented in this section satisfies all four of our Fourier design requirements. In the next sections we will consider methods for optimizing the relatively prime integer values, $s_1, \dots, s_K$, used to construct our $(K, \alpha)_N$-coherent matrices. In what follows we will drop the slight distinction between $(K, \alpha)_N$-coherent and (K, α)-coherent matrices for ease of discussion.

5. Optimizing the (K, α)-Coherent Matrix Construction. Note that K appears as part of a ratio involving α in each statement of Theorem 4. Hence, we will focus on constructing (K, α)-coherent matrices in which K is a constant multiple of α in this section. For a given value of $D \in (1, \infty)$ we can optimize the methods from Section 4 for constructing a $(D\alpha, \alpha)$-coherent matrix with a small number of rows by reformulating the matrix design problem as an optimization problem (see Figure 5.1). In this section we will develop concrete bounds for

Minimize

$$m = \sum_{j=1}^{K_\alpha = \lceil D\alpha \rceil} s_j \qquad (5.1)$$

subject to the following constraints:

I. $s_1 < \cdots < s_K$.

II. $\prod_{j=1}^{\alpha} s_j < N \leq \prod_{j=1}^{\alpha+1} s_j$.

III. $s_1, \cdots, s_K$ are pairwise relatively prime.

FIG. 5.1. *Matrix Design Optimization Problem for given N, D, and α values*

the number of rows, m as a function of D, N, and α, that will appear in any $m \times N$ ($K = D\alpha, \alpha$)-coherent matrix constructed as per Section 4. These bounds will ultimately allow us to cast the matrix optimization problem in Figure 5.1 as an integer linear program in Section 6.

Define $p_0 = 1$ and let p_l be the l^{th} prime natural number. Thus, we have

$$p_0 = 1, p_1 = 2, p_2 = 3, p_3 = 5, p_4 = 7, \ldots \qquad (5.2)$$

Suppose that $S = \{s_1, \ldots, s_K\}$ is a solution to the optimization problem presented in Figure 5.1 for given values of α, D, and N. Let p_{q_S} be the largest prime number appearing in the prime factorization of any element of S. Finally, let

$$q = \max \Big\{ q_S \ \Big| \ S \text{ solves the optimization problem in Figure 5.1} \Big\}.$$

The following lemma bounds q as a function of N, D, and α.

Lemma 6. *Suppose $s_1, s_2, \cdots, s_K$ satisfy all three constraints in Figure 5.1. Set $\tilde{m} = \sum_{j=1}^{K} s_j$. Next, let p_t be the smallest prime number greater than 2 for which*

$$p_t \cdot (K - \alpha - 1) + (K - \alpha - 1)(K - \alpha - 2) + (\alpha + 1) N^{\frac{1}{\alpha+1}} > \tilde{m}$$

holds. Then, $q < t + K - \alpha - 1$.[11]

Proof: Let $s'_1, s'_2, \cdots, s'_K$ be a solution to the optimization problem presented in Figure 5.1. Set $m = \sum_{j=1}^{K} s'_j$. Note that there must exist at least one prime, $p_{\tilde{l}} \in [p_t, p_{t+K-\alpha-1})$, which is not a prime factor of any s'_j value. If no such prime exists, then the sum of $s'_1, s'_2, \cdots, s'_K$ must be

$$m \ \geq \ \sum_{j=0}^{K-\alpha-2} p_{t+j} \ + \ \sum_{j=1}^{\alpha+1} s'_j.$$

[11] Asymptotically, one can always choose α so that t is $O\left(K \cdot \log_K^2 N\right)$. See Section 5.1 for more information regarding the asymptotic behavior of achievable m values.

The second constraint in Figure 5.1 together with the arithmetic-geometric mean inequality tells that we must always have

$$(\alpha+1)\,N^{\frac{1}{\alpha+1}} \;\leq\; (\alpha+1)\cdot\left(\prod_{j=1}^{\alpha+1} s'_j\right)^{\frac{1}{\alpha+1}} \;\leq\; \sum_{j=1}^{\alpha+1} s'_j. \tag{5.3}$$

Furthermore, it is not difficult to see that

$$\sum_{j=0}^{K-\alpha-2} p_{t+j} \;\geq\; \sum_{j=0}^{K-\alpha-2} (p_t+2j) \;\geq\; p_t\cdot(K-\alpha-1) \;+\; (K-\alpha-1)\,(K-\alpha-2) \tag{5.4}$$

since $p_t > 2$. Thus, if every prime in $[p_t, p_{t+K-\alpha-1})$ appears as a prime factor in some s'_j, then

$$m \;\geq\; p_t\cdot(K-\alpha-1) + (K-\alpha-1)\,(K-\alpha-2) + (\alpha+1)\,N^{\frac{1}{\alpha+1}} \;>\; \tilde{m},$$

violating our assumption concerning the optimality of $s'_1, s'_2, \cdots, s'_K$. This proves our claim regarding the existence of at least one prime, $p_{\tilde{l}} \in [p_t, p_{t+K-\alpha-1})$, which is not a prime factor of any s'_j value.

Now suppose that some $s'_{j'}$ contains a prime factor, $p_{l'}$, with $l' \geq t+K-\alpha-1$. Substitute the largest currently unused prime, $p_{\tilde{l}} \in [p_t, p_{t+K-\alpha-1})$, for $p_{l'}$ in the prime factorization of $s'_{j'}$ to obtain a smaller value, $s'_{\tilde{j}}$. If we can show that $s'_1, s'_2, \cdots, s'_K$ with $s'_{\tilde{j}}$ substituted for $s'_{j'}$ still satisfies all three Figure 5.1 constraints after reordering, we will again have a contradiction to the assumed minimality of our original solution. In fact, it is not difficult to see that all constraints other than II above will trivially be satisfied by construction. Furthermore, if $s'_{\tilde{j}} > s'_{\alpha+1}$, then Constraint II will also remain satisfied and we will violate our assumption that the s'_j values originally had a minimal sum.

Finally, the second case where $p_t \leq p_{\tilde{l}} \leq s'_{\tilde{j}} < s'_{\alpha+1}$ could only occur if originally

$$\sum_{j=\alpha+2}^{K} s'_j \geq \sum_{j=1}^{K-\alpha-1} (p_t+2j) = p_t\cdot(K-\alpha-1) + (K-\alpha)\,(K-\alpha-1)\,. \tag{5.5}$$

When combined with Equation (5.3) above, Equation (5.5) reveals that if $s'_{\tilde{j}} < s'_{\alpha+1}$ then we must have originally had

$$\sum_{j=1}^{K} s'_j \geq (\alpha+1)\,N^{\frac{1}{\alpha+1}} + p_t\cdot(K-\alpha-1) + (K-\alpha)\,(K-\alpha-1) > \tilde{m}.$$

However, in this case the assumed minimality of $s'_1, s'_2, \ldots, s'_K$ would again have been violated. □

We will now establish a slightly more refined result than that of Lemma 6.

Lemma 7. *Suppose $s_1, s_2, \cdots, s_K$ satisfy all three constraints in Figure 5.1. Set $\tilde{m} = \sum_{j=1}^{K} s_j$. Let $L = \prod_{i=1}^{v} p_i$ for any desired $v \in \mathbb{N}$, and let $\phi(L) = \prod_{i=1}^{v}(p_i - 1)$. Next, let p_t be the smallest prime number greater than 2 for which*

$$p_t\cdot(K-\alpha-1) + (K-\alpha-1)\,(K-\alpha-2) + \frac{\Big(L-2\phi(L)-2v\Big)(\phi(L)+v)\left\lfloor\frac{K-\alpha-2}{\phi(L)+v}\right\rfloor\Big(\left\lfloor\frac{K-\alpha-2}{\phi(L)+v}\right\rfloor-1\Big)}{2}$$
$$+\Big(L-2\phi(L)-2v\Big)\left\lfloor\frac{K-\alpha-2}{\phi(L)+v}\right\rfloor\left(K-\alpha-1-(\phi(L)+v)\left\lfloor\frac{K-\alpha-2}{\phi(L)+v}\right\rfloor\right) + (\alpha+1)\,N^{\frac{1}{\alpha+1}} > \tilde{m}$$

holds. Then, $q < t + K - \alpha - 1$.

Proof: We will prove this lemma by modifying our proof of Lemma 6. In particular, we will modify formulas (5.4) and (5.5). Note that amongst any L consecutive numbers, there are at most $\phi(L) + v$ prime numbers. Hence, we have that $p_{i+w(\phi(L)+v)} \geq p_i + wL$ for all $w \in \mathbb{N}$. Thus, we may replace formula (5.4) with

$$\begin{aligned}\sum_{j=0}^{K-\alpha-2} p_{t+j} \;\geq\; & p_t \cdot (K-\alpha-1) + (K-\alpha-1)\,(K-\alpha-2) \\ & + \frac{\Big(L - 2\phi(L) - 2v\Big)(\phi(L)+v)\Big\lfloor \frac{K-\alpha-2}{\phi(L)+v} \Big\rfloor\Big(\Big\lfloor \frac{K-\alpha-2}{\phi(L)+v} \Big\rfloor - 1\Big)}{2} \\ & + \Big(L - 2\phi(L) - 2v\Big)\left\lfloor \frac{K-\alpha-2}{\phi(L)+v} \right\rfloor\left(K - \alpha - 1 - (\phi(L)+v)\left\lfloor \frac{K-\alpha-2}{\phi(L)+v} \right\rfloor\right).\end{aligned}$$

Note that amongst any L consecutive numbers, a maximal subset of pairwise relatively prime numbers has at most $\phi(L)+v$ numbers. Hence, we may also replace formula (5.5) by a similar expression. For a more detailed proof see [1]. □

The following corollary of Lemma 6 provides a simple initial upper bound on the largest prime factor that may appear in any solution to the optimization problem presented in Figure 5.1.

Corollary 8. *Let r be such that $\prod_{j=1}^{\alpha} p_{r+j} < N \leq \prod_{j=1}^{\alpha+1} p_{r+j}$, and set $\tilde{m} = \sum_{j=1}^{K} p_{r+j}$. Next, let p_t be the smallest prime larger than 2 for which*

$$p_t \cdot (K-\alpha-1) + (K-\alpha-1)\,(K-\alpha-2) + (\alpha+1)\,N^{\frac{1}{\alpha+1}} > \tilde{m}$$

holds. Then, $q < t + K - \alpha - 1$.

Proof: It is not difficult to see that

$$s_1 = p_{r+1}, s_2 = p_{r+2}, \cdots, s_K = p_{r+K} \tag{5.6}$$

collectively satisfy all three constraints in Figure 5.1. Applying Lemma 6 yields the stated result. □

Similarly, one can obtain the following corollary from Lemma 7.

Corollary 9. *Let r be such that $\prod_{j=1}^{\alpha} p_{r+j} < N \leq \prod_{j=1}^{\alpha+1} p_{r+j}$, and set $\tilde{m} = \sum_{j=1}^{K} p_{r+j}$. Let $L = \prod_{i=1}^{v} p_i$ for any $v \in \mathbb{N}$, and let $\phi(L) = \prod_{i=1}^{v}(p_i - 1)$. Next, let p_t be the smallest prime larger than 2 for which*

$$\begin{aligned}& p_t \cdot (K-\alpha-1) + (K-\alpha-1)\,(K-\alpha-2) + \frac{\Big(L - 2\phi(L) - 2v\Big)(\phi(L)+v)\Big\lfloor \frac{K-\alpha-2}{\phi(L)+v} \Big\rfloor\Big(\Big\lfloor \frac{K-\alpha-2}{\phi(L)+v} \Big\rfloor - 1\Big)}{2} \\ & + \Big(L - 2\phi(L) - 2v\Big)\left\lfloor \frac{K-\alpha-2}{\phi(L)+v} \right\rfloor\left(K - \alpha - 1 - (\phi(L)+v)\left\lfloor \frac{K-\alpha-2}{\phi(L)+v} \right\rfloor\right) + (\alpha+1)\,N^{\frac{1}{\alpha+1}} > \tilde{m}\end{aligned}$$

holds. Then, $q < t + K - \alpha - 1$.

The following lemma provides upper and lower bounds for the members of any valid solution to the optimization problem in Figure 5.1 as functions of N, D, and α. This lemma is critical to the formulation of the optimization problem in Figure 5.1 as an integer linear program in Section 6.

Lemma 10. *The following bounds hold for any valid solution, $S = \{s_1, s_2, \ldots, s_K\}$, to the optimization problem in Figure 5.1:*

1. $s_1 < \cdots < s_K$.
2. $s_1 \geq 2, s_2 \geq 3, \ldots, s_K \geq p_K$.
3. $s_1 < N^{\frac{1}{\alpha}}$ *and* $s_{\alpha+1} > N^{\frac{1}{\alpha+1}}$.
4. *Let $t \in \mathbb{N}$ be defined as in Lemma 6, Lemma 7, Corollary 8, or Corollary 9. Then,* $s_K < p_{t+K-\alpha-1}$.

Proof: Assertion (1) is a restatement of Constraint I in Figure 5.1. The second assertion follows immediately from the fact that the ordered s_j values must be pairwise relatively prime (i.e., Constraint III). The third assertion follows easily from Constraint II. Assertion (4) follows from an argument analogous to the proof of Lemma 6. That is, if $s_K \geq p_{t+K-\alpha-1}$, then we may substitute s_K with the largest prime in $[p_t, p_{t+K-\alpha-1})$ which is not currently a prime factor of $s_1, \ldots, s_K$ and thereby derive a contradiction. □

The following lemmas provide concrete lower bound for m in terms of N, D, and α (see Equation (5.1) in Figure 5.1). These lemmas will ultimately allow us to judge the possible performance of any solution to our optimization problem based solely on the value of α whenever N and D are fixed.

Lemma 11. *Any solution to the optimization problem in Figure 5.1 must have*

$$m \;\geq\; KN^{\frac{1}{\alpha+1}} + (K-\alpha)\,(K-\alpha-1)\,.$$

Proof: We know that $s_{\alpha+2} > s_{\alpha+1} > N^{\frac{1}{\alpha+1}}$ from Lemma 10. Hence, we can see that

$$\sum_{j=\alpha+2}^{K} s_j \;\geq\; \sum_{j=1}^{K-\alpha-1} \left(N^{\frac{1}{\alpha+1}} + 2j\right) \geq (K-\alpha-1)\,N^{\frac{1}{\alpha+1}} + (K-\alpha)\,(K-\alpha-1)\,.$$

Combining this lower bound with Equation (5.3) proves the lemma. □

Corollary 12. *Let $L = \prod_{i=1}^{v} p_i$ for any desired $v \in \mathbb{N}$, and let $\phi(L) = \prod_{i=1}^{v}(p_i - 1)$. Any solution to the optimization problem in Figure 5.1 must have*

$$m \;>\; KN^{\frac{1}{\alpha+1}} + (K-\alpha)\,(K-\alpha-1) + \frac{\Big(L - 2\phi(L) - 2v\Big)(\phi(L)+v)\left\lfloor \frac{K-\alpha-2}{\phi(L)+v} \right\rfloor\left(\left\lfloor \frac{K-\alpha-2}{\phi(L)+v} \right\rfloor - 1\right)}{2}$$
$$+ \Big(L - 2\phi(L) - 2v\Big)\left\lfloor \frac{K-\alpha-2}{\phi(L)+v} \right\rfloor\left(K-\alpha-1-(\phi(L)+v)\left\lfloor \frac{K-\alpha-2}{\phi(L)+v} \right\rfloor\right).$$

Proof: See [1]. □

In the next section we investigate asymptotic bounds of m in terms of D and N. This will, among other things, allow us to judge the quality of our matrices with respect to the lower bound in part 6 of Theorem 4.

5.1. Asymptotic Upper and Lower Bounds. In this section we will utilize standard Vinogradov notation for the sake of readability.[12] We begin by proving an asymptotic lower bound for the number of rows in any (K,α)-coherent matrix created as per Section 4. Recall

[12] Let $g, f : \mathbb{R}^+ \to \mathbb{R}^+$. In Vinogradov notion "$f(x) \ll g(x)$" is equivalent to $f(x) = O(g(x))$, "$f(x) \gg g(x)$" is equivalent to $g(x) \ll f(x)$, and "$f(x) \sim g(x)$" means that $\lim_{x\to\infty} f(x)/g(x) = 1$.

that we have fixed K to be a multiple of α so that $K = K_\alpha = \lceil D\alpha \rceil$ for some $D \in (1, \infty)$. We have the following lower bound for m as a function of D and N.

Lemma 13. *Suppose that $2 \leq D \leq N^{1-\tau}$, where $\tau > 0$ is some fixed constant. For any solution to the optimization problem in Figure 5.1, where α can freely be chosen, one has*

$$m \gg \frac{D^2(\log N)^2}{\log(D \log N)}$$

for sufficiently large values of $D \log N$.

Proof: Let $Q = D \log N$. Suppose that $S = \{s_1, \dots s_K\}$ is a solution to the optimization problem in Figure 5.1, where α can be freely chosen and $K = K_\alpha = \lceil D\alpha \rceil$. By Constraints (I) and (II) in Figure 5.1,

$$\sum_{i=1}^{K} \log s_i > \left\lfloor \frac{D\alpha}{\alpha + 1} \right\rfloor \log N \geq \frac{Q}{4}.$$

Let $q = q_S$ be the largest natural number such that $p_q \mid \prod_{i=1}^{K} s_i$. For $1 \leq i \leq q$, let w_i be the integer such that $p_i^{w_i} \| \prod_{i=1}^{K} s_i$. Since $a + b \leq ab$ for $a, b \in \mathbb{N}$, it follows from Lemma 14 that

$$\sum_{i=1}^{q} p_i^{w_i} \leq \sum_{i=1}^{K} s_i \ll \frac{Q^2}{\log Q}$$

as $Q \to \infty$. This implies that for $1 \leq i \leq q$, we have $p_i^{w_i} \leq \frac{CQ^2}{\log Q}$, where C is some absolute positive constant. Therefore,

$$\sum_{p_i \leq \sqrt{\frac{CQ^2}{\log Q}}} w_i \log p_i \leq \sum_{w=2}^{\infty} \sum_{p \leq \left(\frac{CQ^2}{\log Q}\right)^{1/w}} w \log p \ll \sum_{2 \leq w \leq \frac{\log(CQ^2) - \log\log Q}{\log 2}} \frac{wQ^{2/w}}{(\log Q)^{1/w}} \ll \frac{Q}{\sqrt{\log Q}}.$$

Since $\sum_{i=1}^{q} w_i \log p_i = \sum_{i=1}^{K} \log s_i \geq \frac{Q}{4}$, we have that

$$\sum_{p_i > \sqrt{\frac{CQ^2}{\log Q}}} w_i \log p_i \gg Q$$

for sufficiently large values of Q. Let

$$W = \sum_{p_i > \sqrt{\frac{CQ^2}{\log Q}}} w_i.$$

We have that $w_i \in \{0, 1\}$ when $p_i > \sqrt{\frac{CQ^2}{\log Q}}$. Also, whenever $w_i \geq 1$, it follows that $\log p_i \leq \log \frac{CQ^2}{\log Q} \ll \log Q$. Thus, for sufficiently large values of Q, we have $W \geq \frac{C'Q}{\log Q}$, where C' is some absolute positive constant. By the Prime Number Theorem,

$$\sum_{i=1}^{K} s_i \geq \sum_{i=1}^{q} p_i^{w_i} \geq \sum_{i \leq \frac{C'Q}{\log Q}} p_i \gg \sum_{i \leq \frac{C'Q}{\log Q}} i \log i \gg \frac{Q^2}{\log Q} = \frac{D^2(\log N)^2}{\log(D \log N)}$$

for sufficiently large values of $Q = D\log N$. □

Part 6 of Theorem 4 informs us that m must be $\Omega\left(D^2\log_D N\right)$ for any $m\times N$ $(\lceil D\alpha\rceil, \alpha)$-coherent matrix. On the other hand, Lemma 13 above tells us that any $m \times N$ (K, α)-coherent matrix constructed via Section 4 must have $m = \Omega\left(\frac{D^2(\log N)^2}{\log(D\log N)}\right)$. Note that the lower bounds for matrices constructed as per Section 4 are worse by approximately a factor of $\log N$. This is probably an indication that the $(K = \lceil D\alpha\rceil, \alpha)$-coherent matrix construction in Section 4 is suboptimal. Certainly suboptimality of the construction in Section 4 would not be surprising given that the construction is addressing a more constrained design problem (i.e., we demand small Fourier sampling requirements).

Next we show that the asymptotically best main term for m in the optimization problem in Figure 5.1 can be obtained by taking each s_j to be a prime. This proves that the asymptotic lower bound given in Lemma 13 is tight.

Lemma 14. *Suppose that $2 \leq D \leq N^{1-\tau}$, where $\tau > 0$ is some fixed constant. If we are able to select the value for α, the optimization problem in Figure 5.1 can be solved by taking the s_j to be primes in such a way that guarantees that*

$$m \ll \frac{D^2(\log N)^2}{\log(D\log N)}$$

as $D\log N \to \infty$.

Proof: Let $Q = D\log N$. Since we restrict to the case that $1 \leq D \leq N^{1-\tau}$, it follows that as $Q \to \infty$, we also have that $N \to \infty$. Let $r = \max\left(\left\lceil \frac{Q}{\log Q}\right\rceil, 9\right)$. Note that $\log r > 2$. Also, by the Prime Number Theorem, $p_{r+1} \sim Q \leq N^{1-\tau}\log N$ as $Q \to \infty$. We will assume that Q is large enough that $p_{r+1} < N$. Choose $\alpha \in \mathbb{N}$ such that $\prod_{j=1}^{\alpha} p_{r+j} < N \leq \prod_{j=1}^{\alpha+1} p_{r+j}$. For $1 \leq i \leq K_\alpha = \lceil D\alpha\rceil$, let $s_i = p_{r+i}$. Note that our elements s_i already satisfy the conditions in Figure 5.1. We are left to establish a bound on α and then estimate $\sum_{i=1}^{K_\alpha} s_i$.

Note that $\alpha \leq \beta$ whenever $\prod_{j=1}^{\beta+1} p_{r+j} \geq N$, which is equivalent to $\sum_{j=1}^{\beta+1}\log p_{r+j} \geq \log N$. Let $\beta = \left\lceil \frac{2\log N}{\log r}\right\rceil$. We have that $p_k \geq k$ for $k \geq 1$. Hence, we have that

$$\begin{aligned}\sum_{i=1}^{\beta+1}\log p_{r+i} \geq \sum_{i=1}^{\beta+1}\log(r+i) &\geq \int_r^{r+\beta+1}\log x\,dx\\ &= (\beta+1)(\log(r+\beta+1)-1) + r\log\left(1+\frac{\beta+1}{r}\right)\\ &> (\beta+1)(\log(r+\beta+1)-1) > \beta(\log r - 1) > \frac{2\log N}{\log r}\cdot\frac{\log r}{2} = \log N.\end{aligned}$$

Note that as $Q \to \infty$, $r + K_\beta \ll \frac{Q}{\log Q}$. Thus, since $\alpha \leq \beta$, we have by [29, Lemma 6] and the Prime Number Theorem that

$$\sum_{i=1}^{K_\alpha} p_{r+i} \leq \sum_{i=1}^{K_\beta} p_{r+i} \leq C_1\frac{p_{r+K_\beta}^2}{\log p_{r+K_\beta}} \leq C_2\frac{\left((r+K_\beta)\log(r+K_\beta)\right)^2}{\log\left((r+K_\beta)\log(r+K_\beta)\right)},$$

for some absolute constants C_1 and C_2. As $Q \to \infty$,

$$m \ll \frac{\left((r+K_\beta)\log(r+K_\beta)\right)^2}{\log\left((r+K_\beta)\log(r+K_\beta)\right)} \ll \frac{Q^2}{\log Q} = \frac{D^2(\log N)^2}{\log(D\log N)}.$$ □

Although Lemma 14 shows that simply using primes for our s_j values is asymptotically optimal, it is important to note that the convergence of such primes-only solutions to the optimal value as $D \log N \to \infty$ is likely very slow. For real world values of N and D the more general criteria that the s_j values be pairwise relatively prime can produce significantly smaller m values. This is demonstrated empirically in Section 7. However, Lemma 14 also formally justifies the idea that the s_j values can be restricted to smaller subsets of relatively prime integers (e.g., the prime numbers) before solving the optimization problem in Figure 5.1 without changing the asymptotic performance of the generated solutions. This idea can help make the (approximate) solution of the optimization problem in Figure 5.1 more computationally tractable in practice.

6. Formulation of the Matrix Design Problem as an Integer Linear Program. In this section the optimization problem in Figure 5.1 is formulated as an Integer Linear Program (ILP). This allows the problem to be solved via existing optimization techniques for ILPs which are guaranteed to find an optimal solution whenever one exists (e.g., see [35, 40]). These methods can work well in practice despite the fact that their worst case computational complexities scale exponentially with the number of ILP variables (i.e., roughly like $O\left(2^{K \log(K \log K)}\right)$ in our worst case setting). Furthermore, many of these methods work by producing approximate solutions which progressively converge toward the optimal solution over time. Hence, approximate solutions with accompanying error bounds can be discovered even when the true optimal solution can not be found quickly.

To formulate the problem as an ILP, we define $K = K_\alpha = \lceil D\alpha \rceil$ and $B = p_{t+K-\alpha-1}$ as in Part 4 of Lemma 10. Let $s_{j,i} \in \{0,1\}$ for $j \in [1,K] \cap \mathbb{N}$ and $i \in [1,B] \cap \mathbb{N}$. We then let $s_j = \displaystyle\sum_{i=1}^{B} s_{j,i} \cdot i$ and, for $k \in [1, t+K-\alpha-1] \cap \mathbb{N}$, define

$$\delta_{k,i} = \begin{cases} 1 & \text{if } p_k \mid i \\ 0 & \text{otherwise} \end{cases}. \tag{6.1}$$

Then, for a given α, we can minimize Equation (5.1) by minimizing

$$m = \sum_{j=1}^{K} s_j = \sum_{j=1}^{K} \sum_{i=1}^{B} s_{j,i} \cdot i \tag{6.2}$$

subject to the following linear constraints:

1. $\sum_{i=1}^{B} s_{j,i} = 1$ for all $j \in [1,K] \cap \mathbb{N}$.
2. $s_{j,i} \in \{0,1\}$ for all $j \in [1,K] \cap \mathbb{N}$ and $i \in [1,B] \cap \mathbb{N}$.
3. $\sum_{i=1}^{B} (i \cdot s_{j+1,i} - i \cdot s_{j,i}) \geq 1$ for all $j \in [1, K-1] \cap \mathbb{N}$.
4. $\sum_{j=1}^{\alpha} \sum_{i=1}^{B} s_{j,i} \cdot \ln i < \ln N \leq \sum_{j=1}^{\alpha+1} \sum_{i=1}^{B} s_{j,i} \cdot \ln i$.
5. $s_{j,i} = 0$ for all $j \in [1,K] \cap \mathbb{N}$ and $i \in [1, p_i - 1] \cap \mathbb{N}$.
6. $\sum_{j=1}^{K} \sum_{i=1}^{B} \delta_{k,i} \cdot s_{j,i} \leq 1$ for all $k \in [1, t+K-\alpha-1] \cap \mathbb{N}$.

The first and second constraint together state that for each j, $s_{j,i}$ is non-zero for exactly one value of $i \in [1,B] \cap \mathbb{N}$, implying that $s_j = i$. This in turn, by the third constraint, implies that $s_1 < s_2 < \cdots < s_K$, which is Constraint I in Figure 5.1. Upon applying the natural logarithm in Constraint II in Figure 5.1 to convert a nonlinear constraint to a linear constraint, one obtains something equivalent to our fourth constraint above. The fifth constraint above simply forces $s_j \geq p_j$, which will be true for any solution to the optimization problem in Figure 5.1. The last constraint ensures that $s_1, \ldots, s_K$ are pairwise relatively prime, which is Constraint III in Figure 5.1. Hence, the optimization problem in this section is equivalent to the optimization problem in Figure 5.1.

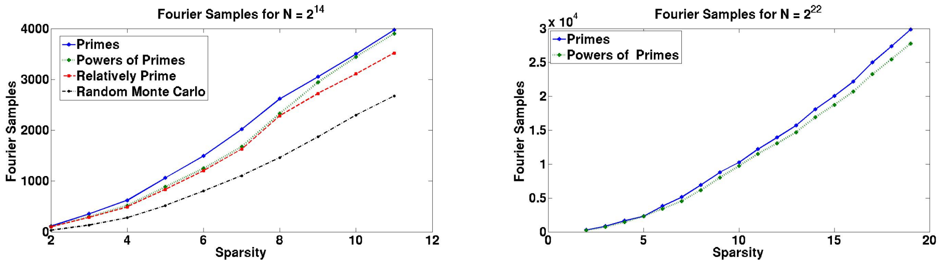

FIG. 7.1. ***On the Left:*** *The minimal Fourier sampling requirements, $m - K_\alpha + 1$ minimized over all feasible $\alpha \in [1, 14] \cap \mathbb{N}$, for any possible $m \times 2^{14}$ $\left(K = \left\lceil \frac{(k-1)\alpha}{\epsilon} \right\rceil, \alpha\right)$-coherent matrix constructed via Section 4. Here ϵ was fixed to be $4/\left(6 + \sqrt{7}\right) \approx 0.463$, and the sparsity parameter, k, was varied between 2 and 11.* ***On the Right:*** *The minimal Fourier sampling requirements, $m - K_\alpha + 1$ minimized over all feasible $\alpha \in [1, 22] \cap \mathbb{N}$, for any possible $m \times 2^{22}$ $\left(K = \left\lceil \frac{(k-1)\alpha}{\epsilon} \right\rceil, \alpha\right)$-coherent matrix constructed via Section 4. Here ϵ was again fixed to be $4/\left(6 + \sqrt{7}\right) \approx 0.463$, and the sparsity parameter, k, was varied between 2 and 19.*

7. Numerical Experiments. In this section we investigate the optimal Fourier sampling requirements related to $m \times N$ $(K = \lceil D\alpha \rceil, \alpha)$-coherent matrices, optimized over the α parameter, for several values of D and N. This is done for given values of $D \in (1, \infty)$ and $N \in \mathbb{N}$ by solving the optimization problem in Figure 5.1 via the ILP presented in Section 6 for all feasible values of $\alpha \in \left[1, \log_2 N\right] \cap \mathbb{N}$.[13] The solution yielding the smallest Fourier sampling requirement, $m - K_\alpha + 1$ from Lemma 5, for the given D and N values (minimized over all α values) is the one reported for experiments in this section. Each ILP was solved with IBM ILOG OPL-CPLEX with parameters generated using Microsoft Visual Studio. Examples of the actual files ran can be downloaded from the contact author's website.[14]

In order to make our numerical experiments more meaningful we computed optimal incoherent matrices which also have the RIP$_2$ (see part 3 of Theorem 4). Hence, we set $D = \frac{k-1}{\epsilon}$ for a given sparsity value $k \in [1, N] \cap \mathbb{N}$ and $\epsilon \in (0, 1)$. In all experiments the value of ϵ was fixed to be slightly less than $3/\left(4 + \sqrt{6}\right) \approx 0.465$ which ensures that l^1-minimization can be utilized with the produced RIP$_2$ matrices for accurate Fourier approximation (e.g., see Theorem 2.7 in [37]).

Three variants of the optimization problem in Figure 5.1 were solved in order to determine the minimal Fourier sampling requirements associated with three types of $\left(\left\lceil \frac{(k-1)\alpha}{\epsilon} \right\rceil, \alpha\right)$-coherent matrices created via Section 4. These three types include the:

1. **Relatively Prime** optimization problem exactly as stated in Figure 5.1 and reformulated in Section 6.
2. **Powers of Primes** optimization problem. Here the s_j values are further restricted to each be a power of a different prime number.
3. **Primes** optimization problem. Here each s_j value is further restricted to simply be a prime number.

These different variants allow some trade off between computational complexity and the minimality of the generated incoherent matrices. See Figure 7.1 for a comparison of the solutions to these optimization problems for two example values of N.

In creating the solutions graphed in Figure 7.1 computer memory was the primary constraining factor. For each of the two values of N the sparsity, k, was increased until computer

[13] It is important to note that many values of α can be disqualified as optimal without solving an ILP by comparing previous solutions to the lower bounds given in Lemma 11 and Corollary 12.

[14] http://www.math.duke.edu/~markiwen/DukePage/code.htm

memory began to run out during the solution of one of the required ILPs.[15] All ILPs which ran to completion did so in less than 90 minutes (most finishing in a few minutes or less). Not surprisingly, the relatively prime solutions always produce smaller Fourier sampling requirements than the more restricted powers of primes solutions, with the tradeoff being that they are generally more difficult to solve. Similarly, the powers of primes solutions always led to smaller Fourier sampling requirements than the even more restricted primes solutions.

For the sake of comparison, the left plot in Figure 7.1 also includes Fourier sampling results for $\text{RIP}_2(k,\epsilon < 0.465)$ matrices created via random sampling based incoherence arguments for each sparsity value. These random Fourier sampling requirements were calculated by choosing rows from a $2^{14} \times 2^{14}$ inverse DFT matrix, $\mathcal{F}^{-1}$, uniformly at random without replacement. After each row was selected, the μ-coherence of the submatrix formed by the currently selected rows was calculated (see Definition 3). As soon as the coherence became small enough that Theorem 3 guaranteed that the matrix would have the $\text{RIP}_2(k,\epsilon < 0.465)$ for the given value of k, the total number of inverse DFT rows selected up to that point was recorded as a trial Fourier sampling value. This entire process was repeated 100 times for each value of k. The smallest Fourier sampling value achieved out of these 100 trials was then reported for each sparsity k in the left plot of Figure 7.1.

Looking at the plot of the left in Figure 7.1 we can see that the randomly selected submatrices guaranteed to have the RIP_2 require fewer Fourier samples than the deterministic matrices constructed herein. Hence, if Fourier sampling complexity is one's primary concern, randomized matrix design techniques should be utilized. However, it is important to note that such randomly constructed Fourier matrices cannot currently be utilized in combination with $o(N)$-time Fourier approximation algorithms. Our deterministic incoherent matrices, on the other hand, have associated sublinear-time approximation algorithms (see the first part of Theorem 4).

Finally, we conclude this paper by noting that heuristic solutions methods can almost certainly be developed for solving the optimization problem in Figure 5.1. Such methods are often successful at decreasing memory usage and computation time while still producing near-optimal results. We leave further consideration of such approaches as future work.

REFERENCES

[1] J. Bailey, M. Iwen, and C. Spencer. On the design of deterministic matrices for fast recovery of fourier compressible functions. *Arxiv preprint arXiv:1105.6138*, 2011.

[2] R. Baraniuk, M. Davenport, R. DeVore, and M. Wakin. A simple proof of the restricted isometry property for random matrices. *Constructive Approximation*, 28(3):253–263, 2008.

[3] R. Berinde, A. C. Gilbert, P. Indyk, H. Karloff, and M. J. Strauss. Combining geometry and combinatorics: a unified approach to sparse signal recovery. *Allerton Conference on Communication, Control and Computing*, 2008.

[4] R. Berinde, A. C. Gilbert, P. Indyk, H. Karloff, and M. J. Strauss. Combining geometry and combinatorics: a unified approach to sparse signal recovery. *arXiv:0804.4666*, 2008.

[5] T. Blumensath and M. E. Davies. Iterative hard thresholding for compressed sensing. *Applied and Computational Harmonic Analysis*, 27(3):265 – 274, 2009.

[6] J. Bourgain, S. Dilworth, K. Ford, S. Konyagin, and D. Kutzarova. Explicit constructions of RIP matrices and related problems. *Preprint*, 2010.

[7] C. Ding and T. Feng. A generic construction of complex codebooks meeting the Welch bound. *IEEE Trans. Inf. Theory*, 53(11):4245–4250, 2007.

[8] E. Candes, J. Romberg, and T. Tao. Stable signal recovery from incomplete and inaccurate measurements. *Communications on Pure and Applied Mathematics*, 59(8):1207–1223, 2006.

[9] E. Candes and T. Tao. Decoding by linear programming. *IEEE Trans. on Information Theory*, 51(12):4203–4215, 2005.

[15] A modest desktop computer with an Intel Core i7-920 processor @ 2.67 Ghz and 2.99 GB of RAM was used to solve all ILPs reported on in Figure 7.1.

[10] E. Candes and T. Tao. Near optimal signal recovery from random projections: Universal encoding strategies? *IEEE Trans. on Information Theory*, 2006.

[11] V. Chandar. A negative result concerning explicit matrices with the restricted isometry property. 2008.

[12] S. Chaudhuri and J. Radhakrishnan. Deterministic restrictions in circuit complexity. *ACM Symposium on Theory of Computing (STOC)*, 1996.

[13] A. Cohen, W. Dahmen, and R. DeVore. Compressed Sensing and Best *k*-term Approximation. *Journal of the American Mathematical Society*, 22(1):211–231, January 2008.

[14] G. Cormode and S. Muthukrishnan. Combinatorial algorithms for compressed sensing. *Lecture Notes in Computer Science*, 4056:280–294, 2006.

[15] D. L. Donoho and M. Elad. Optimally sparse representation in general (nonorthogonal) dictinaries via l1 minimization. *Proc. Natl. Acad. Sci.*, 100:2197–2202, 2003.

[16] D. V. Sarwate. Meeting the Welch bound with equality. *Sequences and Their Applications*, pages 79–102, 1999.

[17] R. A. DeVore. Deterministic constructions of compressed sensing matrices. *Journal of Complexity*, 23, August 2007.

[18] D. L. Donoho and P. B. Stark. Uncertainty Principles and Signal Recovery. *SIAM J. Appl. Math*, 49(3):906 – 931, 1989.

[19] D. Z. Du and F. K. Hwang. *Combinatorial Group Testing and Its Applications*. World Scientific, 1993.

[20] M. Fornasier and H. Rauhut. Compressive sensing. *Handbook of Mathematical Methods in Imaging*, Springer:187–228, 2011.

[21] A. Gilbert, M. Strauss, J. Tropp, and R. Vershynin. One sketch for all: fast algorithms for compressed sensing. In *Proceedings of the thirty-ninth annual ACM symposium on Theory of computing*, pages 237–246. ACM, 2007.

[22] A. C. Gilbert, M. A. Iwen, and M. J. Strauss. Group Testing and Sparse Signal Recovery. In *42nd Asilomar Conference on Signals, Systems, and Computers, Monterey, CA*, 2008.

[23] V. Guruswami, C. Umans, and S. Vadhan. Unbalanced Expanders and Randomness Extractors from ParvareshVardy Codes. *Journal of the ACM*, 56(4), 2009.

[24] M. A. Herman and T. Strohmer. General Deviants: An Analysis of Perturbations in Compressed Sensing. *IEEE Journal of Selected Topics in Signal Processing: Special Issue on Compressive Sensing*, 2010.

[25] P. Indyk and M. Ruzic. Near-Optimal Sparse Recovery in the L1 norm. In *49th Symposium on Foundations of Computer Science*, 2008.

[26] M. Iwen. Combinatorial Sublinear-Time Fourier Algorithms. *Foundations of Computational Mathematics*, 10(3):303 – 338, 2010.

[27] M. Iwen. Improved Approximation Guarantees for Sublinear-Time Fourier Algorithms. *Submitted*, 2010.

[28] M. A. Iwen. Simple Deterministically Constructible RIP Matrices with Sublinear Fourier Sampling Requirements. In *43rd Annual Conference on Information Sciences and Systems (CISS)*, 2009.

[29] M. A. Iwen and C. V. Spencer. Improved bounds for a deterministic sublinear-time Sparse Fourier Algorithm. In *42nd Annual Conference on Information Sciences and Systems (CISS)*, 2008.

[30] F. Krahmer and R. Ward. New and improved Johnson-Lindenstrauss embeddings via the Restricted Isometry Property. *SIAM J. Math. Anal., to appear*, 2011.

[31] J. L. Massey and T. Mittelholzer. Welch's bound and sequence sets for code-division multiple-access systems. *Sequences II: Methods in Communication, Security, and Computer Science*, R. Capocelli, A. De Santis, and U. Vaccoro Eds., 1993.

[32] D. Needell and J. A. Tropp. Cosamp: Iterative signal recovery from incomplete and inaccurate samples. *Applied and Computational Harmonic Analysis*, 26(3), 2008.

[33] D. Needell and R. Vershynin. Uniform uncertainty principle and signal recovery via regularized orthogonal matching pursuit. *Foundations of Computational Mathematics*, 9:317–334, 2009.

[34] D. Needell and R. Vershynin. Signal recovery from incomplete and inaccurate measurements via regularized orthogonal matching pursuit. *IEEE Journal of Selected Topics in Signal Processing*, to appear.

[35] G. Nemhauser and L. Wolsey. *Integer and combinatorial optimization*, volume 18. Wiley New York, 1988.

[36] E. Porat and A. Rothschild. Explicit non-adaptive combinatorial group testing schemes. *ICALP*, 2008.

[37] H. Rauhut. Compressive sensing and structured random matrices. *Theoretical Foundations and Numerical Methods for Sparse Recovery*, vol. 9 in Radon Series Comp. Appl. Math.:1–92, deGruyter, 2010.

[38] M. Rudelson and R. Vershynin. Sparse reconstruction by convex relaxation: Fourier and Gaussian measurements. In *40th Annual Conference on Information Sciences and Systems (CISS)*, 2006.

[39] M. Rudelson and R. Vershynin. On sparse reconstruction from fourier and gaussian measurements. *Comm. Pure Appl. Math.*, 61:1025–1045, 2008.

[40] A. Schrijver. *Theory of linear and integer programming*. John Wiley & Sons Inc, 1998.

[41] T. Strohmer and R. Heath. Grassmanian frames with applications to coding and communication. *Appl. Comp. Harm. Anal.*, 14(3), May 2003.

[42] L. R. Welch. Lower bounds on the maximum cross correlation of signals. *IEEE Trans. Inf. Theory*, 20(3):397–399, 1974.

Appendix A. Proof of Lemma 2.

We will prove a slightly more general variant of Lemma 2. Below we will work with $(K, c_{\min}, \alpha)$-coherent matrices.

Definition 6. *Let $K \in [1, N] \cap \mathbb{N}$ and $c_{\min}, \alpha \in \mathbb{R}^{+}$. An $m \times N$ positive real matrix, $\mathcal{M} \in [0, \infty)^{m \times N}$, is called $(K, c_{\min}, \alpha)$-coherent if both of the following properties hold:*

1. *Every column of $\mathcal{M}$ contains at least K nonzero entries.*
2. *All nonzero entries are at least as large as $c_{\min}$.*
3. *For all $j, l \in [0, N)$ with $j \neq l$, the associated columns, $\mathcal{M}_{\cdot,j}$ and $\mathcal{M}_{\cdot,l} \in [0, \infty)^m$, have $\mathcal{M}_{\cdot,j} \cdot \mathcal{M}_{\cdot,l} \leq \alpha$.*

Clearly, any (K, α)-coherent matrix will also be $(K, 1, \alpha)$-coherent. Other examples of $(K, c_{\min}, \alpha)$-coherent matrices include "corrupted" or "noisy" (K, α)-coherent matrices, as well as matrices whose columns are spherical code words from the first orthant of $\mathbb{R}^m$. In what follows, $\mathcal{M} \in [0, \infty)^{m \times N}$ will always refer to a given $m \times N$ $(K, c_{\min}, \alpha)$-coherent matrix. Let $n \in [0, N) \cap \mathbb{N}$. We define $\mathcal{M}(K, n)$ to be the $K \times N$ matrix created by selecting the K rows of $\mathcal{M}$ with the largest entries in its n^{th} column. Furthermore, we define $\mathcal{M}'(K, n)$ to be the $K \times (N-1)$ matrix created by deleting the n^{th} column of $\mathcal{M}(K, n)$. Thus, if

$$\mathcal{M}_{j_1,n} \ \geq \ \mathcal{M}_{j_2,n} \ \geq \ \ldots \ \geq \ \mathcal{M}_{j_m,n}$$

then

$$\mathcal{M}(K, n) = \begin{pmatrix} \mathcal{M}_{j_1} \\ \mathcal{M}_{j_2} \\ \vdots \\ \mathcal{M}_{j_K} \end{pmatrix} \tag{A.1}$$

and

$$\mathcal{M}'(K, n) = \begin{pmatrix} \mathcal{M}_{j_1,1} & \mathcal{M}_{j_1,2} & \ldots & \mathcal{M}_{j_1,n-1} & \mathcal{M}_{j_1,n+1} & \ldots & \mathcal{M}_{j_1,N} \\ \mathcal{M}_{j_2,1} & \mathcal{M}_{j_2,2} & \ldots & \mathcal{M}_{j_2,n-1} & \mathcal{M}_{j_2,n+1} & \ldots & \mathcal{M}_{j_2,N} \\ & & & \vdots & & & \\ \mathcal{M}_{j_K,1} & \mathcal{M}_{j_K,2} & \ldots & \mathcal{M}_{j_K,n-1} & \mathcal{M}_{j_K,n+1} & \ldots & \mathcal{M}_{j_K,N} \end{pmatrix}. \tag{A.2}$$

The following two lemmas are required for the proof of Lemma 2.

Lemma 15. *Suppose $\mathcal{M}$ is a $(K, c_{\min}, \alpha)$-coherent matrix. Let $n \in [0, N) \cap \mathbb{N}$, $k \in \left[1, \frac{K \cdot c_{\min}^2}{\alpha}\right] \cap \mathbb{N}$, and $\vec{x} \in \mathbb{C}^{N-1}$. Then, at most $\frac{k\alpha}{c_{\min}^2}$ of the K entries of $\mathcal{M}'(K, n) \cdot \vec{x}$ will have magnitude greater than or equal to $\frac{c_{\min}}{k} \cdot \|\vec{x}\|_1$.*

Proof: We have that

$$\left| \left\{ j \ \middle| \ \left| (\mathcal{M}'(K, n) \cdot \vec{x})_j \right| \geq \frac{c_{\min} \cdot \|\vec{x}\|_1}{k} \right\} \right| \leq \frac{k \cdot \left\| \mathcal{M}'(K, n) \cdot \vec{x} \right\|_1}{c_{\min} \cdot \|\vec{x}\|_1} \leq \frac{k}{c_{\min}} \cdot \|\mathcal{M}'(K, n)\|_1.$$

Focusing now on $\mathcal{M}'(K, n)$, we can see that

$$\begin{aligned} \|\mathcal{M}'(K, n)\|_1 &= \max_{l \in [1, N-1] \cap \mathbb{N}} \left\| (\mathcal{M}'(K, n))_{\cdot,l} \right\|_1 \\ &\leq \max_{l \in [0, N) \cap \mathbb{N} - \{n\}} \left\langle \frac{1}{c_{\min}} \left(\mathcal{M}(K, n) \right)_{\cdot,n}, \ \left(\mathcal{M}(K, n) \right)_{\cdot,l} \right\rangle \leq \frac{\alpha}{c_{\min}}. \end{aligned} \tag{A.3}$$

The result follows. □

Lemma 16. *Suppose $\mathcal{M}$ is a $(K, c_{\min}, \alpha)$-coherent matrix. Let $n \in [0, N) \cap \mathbb{N}$, $\tilde{k} \in \left[1, \frac{K \cdot c_{\min}^2}{\alpha}\right] \cap \mathbb{N}$, $S \subset [0, N) \cap \mathbb{N}$ with $|S| \leq \tilde{k}$, and $\vec{x} \in \mathbb{C}^{N-1}$. Then, $\mathcal{M}'(K, n) \cdot \vec{x}$ and $\mathcal{M}'(K, n) \cdot (\vec{x} - \vec{x}_S)$ will differ in at most $\frac{\tilde{k}\alpha}{c_{\min}^2}$ of their K entries.*

Proof: Let $\vec{\mathbb{1}} \in \mathbb{C}^{N-1}$ be the vector of all ones. We have that

$$\left|\left\{j \;\middle|\; (\mathcal{M}'(K, n) \cdot \vec{x})_j \neq (\mathcal{M}'(K, n) \cdot (\vec{x} - \vec{x}_S))_j\right\}\right| = \left|\left\{j \;\middle|\; (\mathcal{M}'(K, n) \cdot \vec{x}_S)_j \neq 0\right\}\right|$$

which is less than or equal to

$$\left|\left\{j \;\middle|\; \left(\mathcal{M}'(K, n) \cdot \vec{\mathbb{1}}_S\right)_j \geq c_{\min}\right\}\right|$$

since all the nonzero entries of $\mathcal{M}'(K, n)$ are at least as large as $c_{\min}$. Applying Lemma 15 with $\vec{x} = \vec{\mathbb{1}}_S$ and $k = \left\|\vec{\mathbb{1}}_S\right\|_1 = |S|$ finishes the proof. □

By combining the two Lemmas above we are able to bound the accuracy with which we can approximate any entry of an arbitrary complex vector $\vec{x} \in \mathbb{C}^N$ using only the measurements from a $(K, c_{\min}, \alpha)$-coherent matrix. We are now equipped to prove a slight generalization of Lemma 2.

Lemma 17. *Suppose $\mathcal{M}$ is a $(K, c_{\min}, \alpha)$-coherent matrix. Let $n \in [0, N) \cap \mathbb{N}$, $k \in \left[1, \frac{K \cdot c_{\min}^2}{\alpha}\right] \cap \mathbb{N}$, $\epsilon \in (0, 1]$, $c \in [2, \infty) \cap \mathbb{N}$, and $\vec{x} \in \mathbb{C}^N$. If $K > c \cdot (k\alpha / c_{\min}^2 \epsilon)$ then $\frac{(\mathcal{M}(K,n) \cdot \vec{x})_j}{(\mathcal{M}(K,n))_{j,n}}$ will be contained in the interval $\left(x_n - \frac{\epsilon \left\|\vec{x} - \vec{x}_{(k/\epsilon)}^{\text{opt}}\right\|_1}{k},\ x_n + \frac{\epsilon \left\|\vec{x} - \vec{x}_{(k/\epsilon)}^{\text{opt}}\right\|_1}{k}\right)$ for more than $\frac{c-2}{c} \cdot K$ values of $j \in [0, K) \cap \mathbb{N}$.*

Proof: Define $\vec{y} \in \mathbb{C}^{N-1}$ to be $\vec{y} = (x_0, x_1, \ldots, x_{n-1}, x_{n+1}, \ldots, x_{N-1})$. We have

$$\mathcal{M}(K, n) \cdot \vec{x} \ = \ x_n \cdot (\mathcal{M}(K, n))_{\cdot, n} + \mathcal{M}'(K, n) \cdot \vec{y}.$$

Applying Lemma 16 with $\tilde{k} = (k/\epsilon)$ demonstrates that at most $\frac{k\alpha}{\epsilon \cdot c_{\min}^2}$ entries of $\mathcal{M}'(K, n) \cdot \vec{y}$ differ from their corresponding entries in $\mathcal{M}'(K, n) \cdot \left(\vec{y} - \vec{y}_{(k/\epsilon)}^{\text{opt}}\right)$. Of the remaining $K - \frac{k\alpha}{\epsilon \cdot c_{\min}^2}$ entries of $\mathcal{M}'(K, n) \cdot \vec{y}$, at most $\frac{k\alpha}{\epsilon \cdot c_{\min}^2}$ will have magnitudes greater than or equal to $\epsilon \cdot c_{\min} \left\|\vec{y} - \vec{y}_{(k/\epsilon)}^{\text{opt}}\right\|_1 / k$ by Lemma 15. Hence, at least

$$K - 2\left(\frac{k\alpha}{\epsilon \cdot c_{\min}^2}\right) > \frac{c-2}{c} \cdot K$$

entries of $\mathcal{M}'(K, n) \cdot \vec{y}$ will have a magnitude no greater than

$$\frac{\epsilon \cdot c_{\min} \left\|\vec{y} - \vec{y}_{(k/\epsilon)}^{\text{opt}}\right\|_1}{k} \leq \frac{\epsilon \cdot c_{\min} \left\|\vec{x} - \vec{x}_{(k/\epsilon)}^{\text{opt}}\right\|_1}{k}.$$

Therefore, $\frac{(\mathcal{M}(K,n) \cdot \vec{x})_j}{(\mathcal{M}(K,n))_{j,n}}$ will approximate x_n to within the stated accurracy for more than $\frac{c-2}{c} \cdot K$ values $j \in [1, K] \cap \mathbb{N}$. □