Sub-linear Time Stochastic Threshold Group Testing via Sparse-Graph Codes

Amirhossein Reisizadeh
Department of Electrical and
Computer Engineering
UCSB
reisizadeh@ucsb.edu

Pedro Abdalla
Department of Electrical and
Computer Engineering
UCSB
pabdallateixeira@hotmail.com

Ramtin Pedarsani
Department of Electrical and
Computer Engineering
UCSB
ramtin@ece.ucsb.edu

***Abstract*—The group testing problem is to identify a population of K defective items in a set of n items using the results of a small number of measurements or tests. In this paper, we study the stochastic threshold group testing problem where the result of each test is positive if the testing pool contains at least u defective items, and the result is negative if the pool contains at most $\ell < u$ defective items. The result is random if the number of defectives lies in the interval (ℓ, u). We leverage tools and techniques from sparse-graph codes and propose a fast decoding algorithm for stochastic threshold group testing. We consider the asymptotic regime that n gets large, $K = \omega(1)$ grows with n and $K = \mathcal{O}(n^\beta)$ for some constant $0 < \beta < 1$, and $u = o(K)$. In this regime, the proposed algorithm requires $\Theta\left(\sqrt{u}K \log^3 n\right)$ tests and recovers all the K defectives with a vanishing error probability. Moreover, our algorithm has a decoding complexity of $\mathcal{O}\left(u^{3/2} K \log^4 n\right)$. This is the first algorithm that solves the stochastic threshold group testing problem with decoding complexity that grows only linearly in K and poly-logarithmically in n.**

I. Introduction

The classical group testing problem aims to find a set of K defective items in a population of total n items by carrying out tests on subsets of the items. The result of each test is positive if the testing subset contains at least one defective item; and is negative otherwise. Rooting back in WWII [1], group testing applications have been arising in a variety of fields since then, such as biology [2], machine learning [3], computer science [4], data analysis [5], signal processing [6], and the Internet of Things [7]. In group testing problem, the goal is to design algorithms that are able to recover the defective items with few number of tests and low decoding complexity.

A. Our Contributions

In this paper, we consider a variant of the group testing problem, namely *stochastic threshold* group testing. In this setting as well, we look for fast and efficient algorithms to test the population and recover the defective items; however, the result of each test is determined as follows. A test's result is negative, if the number of defective items in the testing pool is no greater than a lower threshold ℓ; is positive, if the number of defective items is no less than an upper threshold u; and is random (negative or positive) if the number of defective items in the testing pool lies in the interval (ℓ, u). Clearly, this problem is reduced to the classical group testing problem for the special case of $\ell = 0, u = 1$. In this paper, we consider the regime that $K \in \mathcal{O}(n^\beta) \cap \omega(1)$ for some constant $0 < \beta < 1$ and $u = o(K)$, and propose an algorithm that requires $\Theta(\sqrt{u}K \log^3 n)$ tests and recovers all the K defectives with high probability as n gets large. Furthermore, we show that the decoding complexity of the algorithm is $\mathcal{O}\left(u^{3/2} K \log^4 n\right)$ that is sub-linear in n and grows only linearly in K. The algorithm is based on designing the tests via a sparse-graph code, where the left nodes of the bipartite graph correspond to the items and the right nodes correspond to bundle of tests.

B. Related Work

The classical group testing problem has been largely studied in the literature, and we refer the readers to [2], [8] for a thorough overview of prior work. Most of the proposed works have a decoding complexity that scales polynomially in n. Among the sub-linear time algorithms, one can mention GROTESQUE proposed in [9] that requires $\Theta(K \log K \log n)$ tests and its decoding complexity is $\mathcal{O}(K(\log n + \log K))$, which is sublinear in n. Another sub-linear time algorithm is SAFFRON proposed [10] that is closer to our framework, and employs sparse-graph codes to design the tests. SAFFRON recovers a close-to-one fraction of the defectives using $\Theta(K \log n)$ tests and decoding complexity of $\Theta(K \log n)$, which is order-optimal.

Threshold group testing was first considered in [11] where each test returns 0, if the testing pool contains at most ℓ defective items; returns 1 if the pool contains at least u defective items; and *arbitrarily* returns 0 or 1 otherwise. The author showed that the defective items can be recovered with $\binom{n}{u}$ tests. [12] showed that the number of required tests can be reduced to $\Theta(K^{u-\ell+1} \log K \log(n/K))$. Later, the number of tests was further improved in [13] to $\Theta(K^{3/2} \log(n/K))$. Authors in [14] proposed to use $\mathcal{O}((K/u)^u (K/(K-u))^{K-u} K \log(n/K))$ tests to recover the defective items with decoding complexity $\mathcal{O}(n^u \log n)$. However, all the mentioned algorithms either do not propose an efficient decoding algorithm, or have decoding complexity that grow polynomially in n. More recently, [15] proposed an efficiently decodable non-adaptive group testing scheme, where the decoding complexity grows polynomially in K, but the complexity is far from the optimal one that grows only linearly in K.

In the stochastic threshold group testing problem, if the number of defective items in the pool is in (ℓ, u), the test *randomly* returns 0 or 1. The authors in [16] considered the stochastic threshold group testing problem, and proposed a non-adaptive scheme that requires $\mathcal{O}(\log(1/\epsilon)K\sqrt{\ell}\log n)$ tests to recover the defectives with error probability ϵ, where $\ell = o(K)$ and achieved a decoding complexity of $\mathcal{O}(n\log n + n\log(1/\epsilon))$ that is not sub-linear.

C. Notation

In this paper, we will use the following notations. For any $n \in \mathbb{N}$, we denote by $[n]$ the set $\{1, \cdots, n\}$. For non-negative functions f and g, we write $f(n) = \mathcal{O}(g(n))$ (or $f \in \mathcal{O}(g)$) if there exist $n_0 \in \mathbb{N}$ and $c > 0$ such that $f(n) \leq cg(n)$ for any $n \geq n_0$; and $f(n) = \Theta(g(n))$ if $f(n) = \mathcal{O}(g(n))$ and $g(n) = \mathcal{O}(f(n))$. Moreover, we denote by $f(n) = o(g(n))$ if $f(n)/g(n) \to 0$ when $n \to \infty$; and $f(n) = \omega(g(n))$ if $g(n) = o(f(n))$. For any permutation vector $\mathbf{s}$ on $[n]$ and any set $D \subseteq [n]$, we let $\mathbf{s}(D) = \{\mathbf{s}(d) : d \in D\}$. We denote by $\mathbf{a} \cdot \mathbf{b}$ the element-wise multiplication of the vectors $\mathbf{a}$ and $\mathbf{b}$. For non-negative integers a, b and and integer threshold u, we denote by $a \vee_u b$ the Boolean OR with threshold u, i.e. $a \vee_u b = 1$ if $a + b \geq u$; and $a \vee_u b = 0$ otherwise.

II. Problem Statement

The group testing (GT) problem aims at finding K defective items out of n items using the results of t tests. Let the binary vector $\mathbf{x} = (x_1, ..., x_n) \in \{0,1\}^n$ denote the n items, where $x_j = 1$ if item j is defective and $x_j = 0$, otherwise. The threshold group testing problem aims to design a measurement matrix $\mathbf{A} \in \{0,1\}^{t \times n}$ where each row $\mathbf{a}_i = (a_{i1}, \cdots, a_{in})$ denotes the items contributing to the ith test; that is, for $i \in [t]$ and $j \in [n]$, $a_{ij} = 1$ if item j contributes to the ith test and $a_{ij} = 0$, otherwise. In stochastic threshold GT, given lower and upper thresholds ℓ and u, the result of the ith measurement on $\mathbf{x}$ is a random binary number which we denote by $\langle \mathbf{a}_i, \mathbf{x} \rangle_\ell^u$ and is obtained as follows:

$$\langle \mathbf{a}_i, \mathbf{x} \rangle_\ell^u = \begin{cases} 0, & \text{if } \sum_{j=1}^{n} a_{ij}x_j \leq \ell, \\ 1, & \text{if } \sum_{j=1}^{n} a_{ij}x_j \geq u, \\ \text{Bern}(1/2), & \text{otherwise.} \end{cases} \quad (1)$$

The results of the t tests can be aggregated in a binary vector $\mathbf{y} \in \{0,1\}^t$, i.e.

$$\mathbf{y} = \mathbf{A} \odot \mathbf{x} = \begin{bmatrix} \langle \mathbf{a}_1, \mathbf{x} \rangle_\ell^u \\ \vdots \\ \langle \mathbf{a}_t, \mathbf{x} \rangle_\ell^u \end{bmatrix}. \quad (2)$$

The operator defined in (1) implies that each test returns 1 if there are at least u defective items in the testing pool; it returns 0 if there are at most ℓ defective items in the testing pool; and the test returns an equally probable random 0 or 1 if the number of defective items in the testing pool lies in the interval (ℓ, u). In this paper, we consider the asymptotic regime that the upper threshold grows $u = o(K)$ as n and $K \ll n$ get large. The goal of this paper is to design a fast and provably efficient algorithm that is able to find all the defective items with high probability as n gets large, and has near-optimal number of measurements, i.e. $t = \mathcal{O}(K\text{polylog}(n))$ and a decoding algorithm that has sub-linear time complexity which grows only linearly in K and poly-logarithmically in n.

III. Main Results

In this section, we propose our algorithm which consists of the design of the measurement matrix and decoding the test results. We further provide the main result of the paper in Theorem 1, and analyze the performance of the proposed algorithm.

Before describing the measurement matrix design, we first explain our approach in dealing with the uncertainty in the test results as defined in (1) and (2). The following lemma shows that by repeating a test $\Theta(\log n)$ times, one can obtain an equivalent but deterministic test result with high probability.

Lemma 1. *By $r = c_r \log n$ repetitions of the stochastic threshold operation (1) with thresholds (ℓ, u), the test result is equivalent to a (deterministic) Boolean OR operation with threshold u, with probability of error no greater than $\frac{1}{n^{c_r}}$, for a positive constant c_r.*

Proof: Let $y^1, \cdots, y^r$ denote the test result $\langle \mathbf{a}, \mathbf{x} \rangle_\ell^u$ for r realizations. Consider $\overline{y} = y^1 \cdots y^r$, the multiplication of the r realizations. Then, $\overline{y}$ differs from the Boolean OR operation with threshold u, i.e. $a_1x_1 \vee_u \cdots \vee_u a_nx_n$ only if $\ell < \sum_{j=1}^n a_jx_j < u$. Hence,

$$\begin{aligned} \Pr\left[\overline{y} \neq a_1x_1 \vee_u \cdots \vee_u a_nx_n\right] &\leq \Pr\left[y^1 \cdots y^r \neq 0\right] \\ &= \left(\frac{1}{2}\right)^{c_r \log n} = \frac{1}{n^{c_r}}. \end{aligned}$$

■

Next, we present our sparse-graph-based measurement design and precisely describe the decoding procedure.

A. Multi-ton Only Algorithm for Threshold Group Testing

We define a random bipartite graph with n left nodes corresponding to the items and M right nodes each associated with a bundle of tests. Each left node is connected to any right node with probability p, independent of other connections. Let $\mathcal{G}(n, M)$ denote the random bipartite graph with associated adjacency matrix $T_\mathcal{G} \in \{0,1\}^{M \times n}$. We further design a binary signature vector of length h for each item that will be explained shortly. Given a *signature matrix* $\mathbf{U} \in \{0,1\}^{h \times n}$, we assign h tests to every right node as follows. For each right node $i \in [M]$, we denote by $\mathbf{t}_i \in \{0,1\}^n$ the ith row of the adjacency matrix $T_\mathcal{G}$. The measurement matrix associated with right node i is defined as $\overline{\mathbf{A}}_i = \mathbf{U}\,\text{diag}(\mathbf{t}_i) \in \{0,1\}^{h \times n}$. Let $\overline{\mathbf{A}} = [\overline{\mathbf{A}}_1; \cdots; \overline{\mathbf{A}}_M] \in \{0,1\}^{m \times n}$ where $m = M \times h$. We define the overall measurement matrix as concatenation of r copies of $\overline{\mathbf{A}}$, that is $\mathbf{A} = [\overline{\mathbf{A}}; \cdots; \overline{\mathbf{A}}] \in \{0,1\}^{t \times n}$. Therefore, our algorithm designs a total of $t = m \times r = M \times h \times r$ tests. We also propose to merge the results of t tests as follows. Let $\mathbf{y} = [\mathbf{y}^1; \cdots; \mathbf{y}^r] \in \{0,1\}^t$ denote the test results as defined in (2). We then let

$$\overline{\mathbf{y}} = \mathbf{y}^1 \cdot \cdots \cdot \mathbf{y}^r \in \{0,1\}^m \quad (3)$$

be the final test results which will be later used to recover the defective items.

We design the signature matrix $\mathbf{U}$ as concatenation of $c_p + 1$

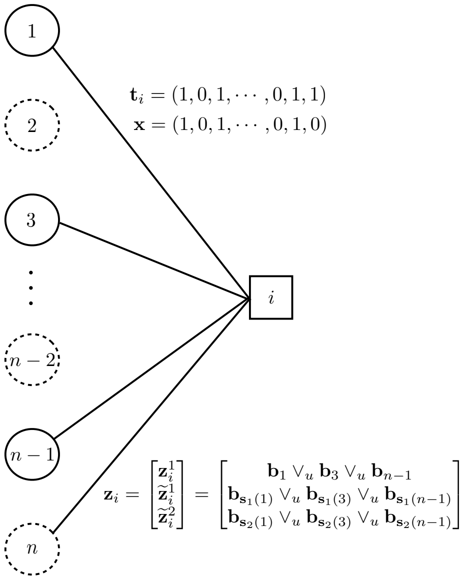
1
2
3
$n-2$
$n-1$
n
i
$\mathbf{t}_i = (1,0,1,\cdots,0,1,1)$
$\mathbf{x} = (1,0,1,\cdots,0,1,0)$
$\mathbf{z}_i = \begin{bmatrix}\mathbf{z}_i^1\\ \widetilde{\mathbf{z}}_i^1\\ \widetilde{\mathbf{z}}_i^2\end{bmatrix} = \begin{bmatrix}\mathbf{b}_1 \vee_u \mathbf{b}_3 \vee_u \mathbf{b}_{n-1}\\ \mathbf{b}_{\mathbf{s}_1(1)} \vee_u \mathbf{b}_{\mathbf{s}_1(3)} \vee_u \mathbf{b}_{\mathbf{s}_1(n-1)}\\ \mathbf{b}_{\mathbf{s}_2(1)} \vee_u \mathbf{b}_{\mathbf{s}_2(3)} \vee_u \mathbf{b}_{\mathbf{s}_2(n-1)}\end{bmatrix}$

Fig. 1: An illustration of a right node measurement.

blocks, for some constant positive integer c_p as follows:

$$\mathbf{U} = \begin{bmatrix}\mathbf{U}_1\\ \widetilde{\mathbf{U}}_1\\ \vdots\\ \widetilde{\mathbf{U}}_{c_p}\end{bmatrix} = \begin{bmatrix}\mathbf{b}_1 & \mathbf{b}_2 & \dots & \mathbf{b}_n\\ \mathbf{b}_{\mathbf{s}_1(1)} & \mathbf{b}_{\mathbf{s}_1(2)} & \cdots & \mathbf{b}_{\mathbf{s}_1(n)}\\ \vdots & & & \\ \mathbf{b}_{\mathbf{s}_{c_p}(1)} & \mathbf{b}_{\mathbf{s}_{c_p}(2)} & \dots & \mathbf{b}_{\mathbf{s}_{c_p}(n)}\end{bmatrix}. \quad (4)$$

In (4), matrix $\mathbf{U}_1$ is a collection of n random binary non-zero column vectors $\mathbf{b}_1, \mathbf{b}_2, \cdots, \mathbf{b}_n$ each of length $c\log n$ for some constant $c > 0$. Thus, $h = (c_p+1)c\log n$. Any entry of these n vectors is a Bernoulli random variable with parameter $(\frac{1}{2})^{1/u}$, which is drawn independent of other entries. Each of the matrices $\widetilde{\mathbf{U}}_1, \cdots, \widetilde{\mathbf{U}}_{c_p}$ contains the same set of columns as in $\mathbf{U}_1$, but ordered with respect to permutation vectors $\mathbf{s}_1, \cdots, \mathbf{s}_{c_p}$, respectively. Vectors $\mathbf{s}_1, \cdots, \mathbf{s}_{c_p}$ are drawn independently and uniformly at random from the set of all $n!$ permutations of the numbers in $[n]$.

Thus far, we have described the design of the measurement matrix $\mathbf{A}$ based on random sparse-graph codes, which outputs the test results $\overline{\mathbf{y}}$ when applied to $\mathbf{x}$, as defined in (1), (2) and (3). For the sake of notation simplicity, we let $\mathbf{z}_i$ denote the observation vector corresponding to right node $i \in [M]$, i.e.

$$\mathbf{z}_i := \overline{\mathbf{y}}_{(i-1)h+1:ih}. \quad (5)$$

According to the signature matrix in (4), each $\mathbf{z}_i$ consists of $c_p + 1$ sections which we denote by $\mathbf{z}_i = [\mathbf{z}_i^1; \widetilde{\mathbf{z}}_i^1; \cdots; \widetilde{\mathbf{z}}_i^{c_p}]$ for every right node i. Fig. 1 illustrates the measurement of right node i for $c_p = 2$ which is connected to defective items $1, 3, n-1$. As proved in Lemma 1, after repeating the stochastic threshold operation (1) and merging the test results, w.h.p. $\mathbf{z}_i$ can be written as the Boolean OR (with threshold u) of columns of the signature matrix indexed by $1, 3, n-1$.

In the following, we describe our proposed procedure to recover the defective items from the test results. First, we generate the look-up matrix $\mathbf{C} \in \{0,1\}^{c\log n \times \binom{n}{u}}$ which will be used to decode the test results. Each column of $\mathbf{C}$ is the entry-wise multiplication of u different columns of $\mathbf{U}_1 = [\mathbf{b}_1, \mathbf{b}_2, \dots, \mathbf{b}_n]$, i.e. the kth column of $\mathbf{C}$ is $\mathbf{c}_k = \mathbf{b}_{k_1} \cdot \mathbf{b}_{k_2} \cdot \cdots \cdot \mathbf{b}_{k_u}$ for some distinct indices $D_k = \{k_1, \dots, k_u\} \subseteq [n]$. Thus, every column of $\mathbf{C}$ corresponds to a set of u items, and there are a total of $\binom{n}{u}$ columns.

Remark 1. Matrices $\mathbf{U}$ and $\mathbf{C}$ are used for measurement and decoding, respectively. Therefore, one can generate and store them once off-line and repeatedly use them for several test pools. Moreover, columns of $\mathbf{C}$ are ordered (increasingly with respect to their binary representations) to reduce the search complexity as will be discussed later.

Next, we describe the decoding procedure. We first define a u-ton right node as follows. *A right node of the bipartite graph is called a u-ton if it is connected to exactly u defective items.* Our algorithm iterates through each right node i once and tries to match the first section of its test results, i.e. $\mathbf{z}_i^1$ with columns of $\mathbf{C}$. Assume such match exists, i.e. $\mathbf{z}_i^1 = \mathbf{c}_k$ for some k that corresponds to a set of u items D_k. Same procedure is applied to the remaining c_p sections of test results, i.e. $\widetilde{\mathbf{z}}_i^1, \cdots, \widetilde{\mathbf{z}}_i^{c_p}$ to obtain c_p sets of u items $D_{k^1}, \cdots, D_{k^{c_p}}$, respectively, if there exist c_p columns of $\mathbf{C}$ that are equal to $\widetilde{\mathbf{z}}_i^1, \cdots, \widetilde{\mathbf{z}}_i^{c_p}$. We then *check* whether the set of items obtained from $D_k, D_{k^1}, \cdots, D_{k^{c_p}}$ are consistent with permutation vectors $\mathbf{s}_1, \cdots, \mathbf{s}_{c_p}$. More precisely, we declare right node i as u-ton if the c_p check equations hold, that is $D_{k^1} = \mathbf{s}_1(D_k), \cdots, D_{k^{c_p}} = \mathbf{s}_{c_p}(D_k)$. Moreover, the u items in D_k are declared as defectives.

Lemma 2. *Our algorithm recovers any certain u-ton with probability $1 - \mathcal{O}\left(\frac{1}{n^{c-2u}}\right)$. Moreover, for w-tons where $w \neq u$, the algorithm wrongly detects an item as defective with probability no greater than $\left(\frac{u}{n}\right)^{c_p}$.*

Proof: Let right node i be a u-ton connecting to defective items $D_k = \{k_1, \cdots, k_u\}$. First, we show that there exists a column $\mathbf{c}_k$ in $\mathbf{C}$ that matches $\mathbf{z}_i^1$ and corresponds to items D_k. As we showed in Lemma 1, the test result $\mathbf{z}_i^1$ (after repetitions and w.h.p.) can be written as $\mathbf{z}_i^1 = \mathbf{b}_{k_1} \vee_u \cdots \vee_u \mathbf{b}_{k_u}$. In other words, the jth element of $\mathbf{z}_i^1$ is

$$z_{i,j}^1 = \begin{cases} 1, & \text{if } b_{k_1,j} + \cdots + b_{k_u,j} \geq u,\\ 0, & \text{if } b_{k_1,j} + \cdots + b_{k_u,j} < u, \end{cases} \quad (6)$$

for all j. This implies that $z_{i,j}^1 = 1$ if and only if $b_{k_1,j} = \cdots = b_{k_u,j} = 1$. Therefore, we can write $\mathbf{z}_i^1 = \mathbf{b}_{k_1} \cdot \mathbf{b}_{k_2} \cdot \cdots \cdot \mathbf{b}_{k_u}$. This ensures that there exits a column $\mathbf{c}_k$ in $\mathbf{C}$ that matches $\mathbf{z}_i^1$ and corresponds to items D_k.

Now, we prove that there is only one column in $\mathbf{C}$ matching $\mathbf{z}_i^1$, with high probability. Assume that there exists $k' \neq k$ such that $\mathbf{c}_k = \mathbf{c}_{k'}$. Let $D_{k'} = \{k'_1, \cdots, k'_u\} \neq D_k$ denote the items corresponding to $\mathbf{c}_{k'}$. We can write

$$\begin{aligned}\Pr[\mathbf{c}_k = \mathbf{c}_{k'}] &= \prod_{j=1}^{c\log n} \Pr[c_{k,j} = c_{k',j}]\\ &= \left(\Pr[c_{k,1} = c_{k',1}]\right)^{c\log n}\\ &= \left(\Pr\left[b_{k_1,1} \cdots b_{k_u,1} = b_{k'_1,1} \cdots b_{k'_u,1}\right]\right)^{c\log n}\\ &\overset{(a)}{\leq} \left(\frac{1}{2}\right)^{c\log n} = \frac{1}{n^c}. \end{aligned} \quad (7)$$

To prove inequality (a) in (7), assume that $D_{k'}$ and $D_{k'}$ have

v items in common, e.g. $k_1 = k'_1, \cdots, k_v = k'_v$. We can write

$$\begin{aligned}
&\Pr\left[b_{k_1,1}\cdots b_{k_u,1} = b_{k_1,1}\cdots b_{k_v,1} b_{k'_{v+1},1}\cdots b_{k'_u,1}\right]\\
&\quad = \Pr\left[b_{k_{v+1},1}\cdots b_{k_u,1} = b_{k'_{v+1},1}\cdots b_{k'_u,1}\right]\\
&\quad \times \Pr\left[b_{k_1,1}\cdots b_{k_v,1} = 1\right] + \Pr\left[b_{k_1,1}\cdots b_{k_v,1} = 0\right]\\
&\quad \overset{(b)}{=} \left(\frac{1}{2}\right)^{\frac{2(u-v)}{u}} \left(1 - \left(\frac{1}{2}\right)^{\frac{(u-v)}{u}}\right)^2 \left(\frac{1}{2}\right)^{\frac{v}{u}} + 1 - \left(\frac{1}{2}\right)^{\frac{v}{u}}\\
&\quad \overset{(c)}{=} \theta^2\left(1-\theta\right)^2\left(\frac{1}{2\theta}\right) + 1 - \frac{1}{2\theta} =: f(\theta) \overset{(d)}{\leq} \frac{1}{2}.
\end{aligned}$$

In above, (b) follows from the fact that all the random variables $b_{i,j}$ are i.i.d. Bern $\left((\frac{1}{2})^{1/u}\right)$. Letting $\theta = \left(\frac{1}{2}\right)^{(u-v)/u}$ yields (c) and $1/2 \leq \theta < 1$. Finally, (d) follows from the fact that the function $f(\cdot)$ is increasing in $[1/2, 1]$ and $f(1) = 1/2$.

The probability of any two columns in $\mathbf{C}$ to be equal is then bounded by $\left(\binom{\binom{n}{u}}{2}\right)/n^c = \mathcal{O}\left(\frac{1}{n^{c-2u}}\right)$. Putting all pieces together, reading the first section of test results, i.e. $\mathbf{z}_i^1$ from the look-up matrix $\mathbf{C}$ detects defective items D_k with probability at least $1 - \mathcal{O}\left(\frac{1}{n^{c-2u}}\right)$. By the same argument, reading $\widetilde{\mathbf{z}}_i^1, \cdots, \widetilde{\mathbf{z}}_i^{c_p}$ would also result in sets $\mathbf{s}_1\left(D_k\right), \cdots, \mathbf{s}_{c_p}\left(D_k\right)$, respectively with the same probability. Therefore, items in D_k are declared as defectives with probability no less than $1 - \mathcal{O}\left(\frac{1}{n^{c-2u}}\right)$.

On the other hand, assume that right node i is not a u-ton and our algorithm wrongly declares a non-defective as a defective connected to node i. More precisely, let D_k denote the set of u items declared by the algorithm as defectives connected to right node i, where at least one of them, e.g. $k_1 \in D_k$ is not actually defective. Therefore, signature vector indexed by k_1 does not contribute to the first section of the test results, $\mathbf{z}_i^1$. This implies that neither do signature vectors indexed by $\mathbf{s}_1(k_1), \cdots, \mathbf{s}_{c_p}(k_1)$ contribute to the remaining c_p sections of the test results, $\widetilde{\mathbf{z}}_i^1, \cdots, \widetilde{\mathbf{z}}_i^{c_p}$ respectively. It yields that $\mathbf{s}_1(k_1), \cdots, \mathbf{s}_{c_p}(k_1)$ are independent of $D_{k^1}, \cdots, D_{k^{c_p}}$, respectively. Our algorithm wrongly declares all of the items in D as defectives, only if all the check equations $D_{k^1} = \mathbf{s}_1\left(D_k\right), \cdots, D_{k^{c_p}} = \mathbf{s}_{c_p}\left(D_k\right)$ hold. The chance of each to hold is

$$\begin{aligned}
\Pr\left[D_{k_1} = \mathbf{s}_1\left(D_k\right)\right] &\leq \Pr\left[\mathbf{s}_1(k_1) \in D_{k_1}\right]\\
&\leq \sum_{j \in D_{k_1}} \Pr\left[\mathbf{s}_1(k_1) = j\right] \leq \frac{u}{n}.
\end{aligned}$$

The same argument holds for each check equation. Therefore, for a certain right node which is not a u-ton, our algorithm wrongly declares a non-defective item as a defective connected to the right node with probability $\left(\frac{u}{n}\right)^{c_p}$. ■

The following example illustrates the main concepts and ideas of designing the tests and recovering the defective items in the proposed algorithm.

Example 1. Consider a set of $n = 4$ items with $k = 3$ defective items identified by $\mathbf{x} = (1, 1, 0, 1)$, i.e. items $1, 2, 4$ are defective. Consider a bipartite graph with adjacency matrix

$$T_{\mathcal{G}} = \begin{bmatrix} 1 & 1 & 1 & 0\\ 0 & 1 & 0 & 1\\ 1 & 1 & 0 & 1 \end{bmatrix},$$

where each row corresponds to one of the $M = 3$ right nodes. We assign $c \log n = 4$ tests to each right node according to the following signature matrix ($c_p = 2$)

$$\mathbf{U} = \begin{bmatrix} \mathbf{U}_1\\ \widetilde{\mathbf{U}}_1\\ \widetilde{\mathbf{U}}_2 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 0 & 1\\ 1 & 1 & 1 & 1\\ 0 & 1 & 1 & 1\\ 1 & 1 & 1 & 0\\ 0 & 1 & 1 & 1\\ 1 & 1 & 1 & 1\\ 1 & 1 & 0 & 1\\ 1 & 0 & 1 & 1\\ 1 & 0 & 1 & 1\\ 1 & 1 & 1 & 1\\ 0 & 1 & 1 & 1\\ 1 & 1 & 1 & 0 \end{bmatrix} = \left[\mathbf{u}_1, \mathbf{u}_2, \mathbf{u}_3, \mathbf{u}_4\right],$$

with random permutations $\mathbf{s}_1 = (3, 4, 1, 2)$ and $\mathbf{s}_2 = (1, 3, 2, 4)$ and upper threshold $u = 2$. Moreover, the columns of the look up matrix $\mathbf{C}$ can be generated and sorted as follows

$$\mathbf{C} = \begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 1\\ 1 & 1 & 1 & 1 & 1 & 1\\ 0 & 1 & 1 & 0 & 0 & 1\\ 1 & 0 & 1 & 0 & 1 & 0 \end{bmatrix},$$

whose for instance, the first column is the element-wise multiplication of the first and third columns of $\mathbf{U}_1$. According to the measurement defined in (2) and the guarantee provided by Lemma 1, w.h.p. the test result is

$$\overline{\mathbf{y}} = \begin{bmatrix} \mathbf{z}_1\\ \mathbf{z}_2\\ \mathbf{z}_3 \end{bmatrix} = \begin{bmatrix} \mathbf{u}_1 \vee_2 \mathbf{u}_2\\ \mathbf{u}_2 \vee_2 \mathbf{u}_4\\ \mathbf{u}_1 \vee_2 \mathbf{u}_2 \vee_2 \mathbf{u}_4 \end{bmatrix},$$

where the right node measurements are

$$\begin{aligned}
\mathbf{z}_1 &= (1, 1, 0, 1, 0, 1, 1, 0, 0, 1, 0, 1)^\top,\\
\mathbf{z}_2 &= (1, 1, 1, 0, 1, 1, 1, 0, 0, 1, 1, 0)^\top,\\
\mathbf{z}_3 &= (1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1)^\top.
\end{aligned}$$

Reading $\mathbf{z}_1$ from the look up matrix yields that $D = \{1, 2\}$, $D_1 = \{3, 4\}$ and $D_2 = \{1, 3\}$. Since the check equations $D_1 = \mathbf{s}_1\left(D\right)$ and $D_2 = \mathbf{s}_2\left(D\right)$ hold, our algorithm declares right node 1 a 2-ton and $\{1, 2\}$ as defectives connected to it. Similarly, right node 2 is detected as 2-ton with defectives $\{2, 4\}$. There is no match with result of right node 3 in the look up matrix, thus the algorithm terminates and outputs items $\{1, 2, 4\}$ as defectives.

B. Analysis

As the main theorem of the paper, we evaluate the performance of the proposed algorithm in the following.

Theorem 1. *With $t = e(c_p + 1)c_r c_u \alpha \left(\frac{2u+c_u}{\sqrt{u}}\right) K \log^3 n$ tests, our algorithm recovers all of the $K \in \mathcal{O}(n^\beta) \cap \omega(1)$ defective items ($0 < \beta < 1$) with probability $1 - \mathcal{O}\left(K\left(\frac{1}{n^\alpha} + \frac{K \log n}{\sqrt{u} n^{c_u}}\right) + \frac{K \log n}{\sqrt{u} n^{c_p(1-\beta)}} + \frac{\sqrt{u} K \log^2 n}{n^{c_r}}\right)$ for positive constants c_p, c_r, c_u, and α. Moreover, the decoding complexity of the algorithm is $\Theta(u^{3/2} K \log^4 n)$.*

Proof: There are three types of error our algorithm might make, which are

(I) A defective item is not detected.

(II) A non-defective item is wrongly declared as defective.

(III) Test results do not agree with the deterministic threshold operation.

In the following, we bound the probability of each error type.

Firstly, as we described in Section III-A, our algorithm iterates through each right node once and detects u-tons with high probability which was characterized in Lemma 2. Therefore, a defective item remains unidentified at the end of the algorithm only if it is not connected to any u-tons; or it is connected to a u-ton but is missed to detect. To characterize the former, consider the pruned bipartite graph consisting of the K defectives and $M = \frac{1}{\sqrt{u}}e\alpha K\log n$ right nodes for some positive constant $\alpha = \Theta(1)$, where any edge in the bipartite graph exists with probability $p = u/K$ independent of other edges. Hence, the average degree of a right node is $\lambda = Kp = u$. Since $u = o(K)$ and $K = \omega(1)$, Poisson approximation for binomial distribution holds as K gets large. Thus, the probability of a right node being u-ton is $p_u = e^{-\lambda}\frac{\lambda^u}{u!} = e^{-u}\frac{u^u}{u!} \geq \frac{1}{e\sqrt{u}}$. Denoted by ϵ_1 is the probability that a certain defective item k is not connected to any u-ton, which we can write as

$$\begin{aligned}\epsilon_1 &= \sum_{d=0}^{M}(1-p_u)^d\binom{M}{d}p^d(1-p)^{M-d} = (1-pp_u)^M\\ &\leq \left(e^{-pp_u}\right)^M \leq \left(e^{-\frac{u}{K}\cdot\frac{1}{e\sqrt{u}}}\right)^{\frac{1}{\sqrt{u}}e\alpha K\log n} = \frac{1}{n^{\alpha}}. \qquad (8)\end{aligned}$$

On the other hand, Lemma 2 together with the union bound on the $M = \Theta\left(\frac{1}{\sqrt{u}}K\log n\right)$ right nodes shows that if k is connected to a u-ton, then it remains unidentified with probability no greater than $\mathcal{O}\left(\frac{K\log n}{\sqrt{u}n^{c-2u}}\right) = \mathcal{O}\left(\frac{K\log n}{\sqrt{u}n^{c_u}}\right)$, where we picked $c = 2u + c_u$ for a positive tuning parameter $c_u = \Theta(1)$. Together with (8), we have

$$\begin{aligned}\Pr[\mathrm{I}] &\leq K\Pr[k \text{ remains unidentified}]\\ &\leq \mathcal{O}\left(K\left(\frac{1}{n^{\alpha}} + \frac{K\log n}{\sqrt{u}n^{c_u}}\right)\right). \qquad (9)\end{aligned}$$

Secondly, by using the union bound and Lemma 2, we have

$$\Pr[\mathrm{II}] \leq \mathcal{O}\left(M\left(\frac{u}{n}\right)^{c_p}\right) \leq \mathcal{O}\left(\frac{K\log n}{\sqrt{u}n^{c_p(1-\beta)}}\right), \qquad (10)$$

where we used the fact that $u/n \leq K/n = \mathcal{O}(1/n^{1-\beta})$. Finally, using Lemma 1 and union bound, we have

$$\Pr[\mathrm{III}] \leq \frac{Mh}{n^{c_r}} = \mathcal{O}\left(\frac{\frac{c}{\sqrt{u}}K\log^2 n}{n^{c_r}}\right) = \mathcal{O}\left(\frac{\sqrt{u}K\log^2 n}{n^{c_r}}\right). \qquad (11)$$

Therefore, with $t = M \times h \times r = e(c_p + 1)c_r c_u\alpha\left(\frac{2u+c_u}{\sqrt{u}}\right)K\log^3 n$ tests, our algorithm detects all the defectives with the error probability as claimed in the theorem. Note that all types of error have vanishing probability by tuning the constants α, c_u and c_r, and c_p. In particular, type II error probability in (10) is vanishing for proper pick of c_p, e.g. $c_p = 2/(1-\beta)$.

Regarding the decoding complexity, we note that searching the ordered look-up matrix for a match to each of the $\Theta(\sqrt{u}K\log^3 n)$ test results has time complexity $\mathcal{O}(\log\binom{n}{u}) = \mathcal{O}(u\log n)$, which implies an overall decoding complexity of $\mathcal{O}(u^{3/2}K\log^4 n)$. ■

IV. Conclusion

In this paper, we proposed a sub-linear time stochastic threshold group testing algorithm based on random sparse-graph codes. We considered the asymptotic regime that the number of items n gets large, $K \in \mathcal{O}(n^{\beta}) \cap \omega(1)$ for some constant $0 < \beta < 1$, and $u = o(K)$, and proposed a fast and provably efficient algorithm that has sample complexity $\Theta(\sqrt{u}K\log^3 n)$ tests and recovers all the K defectives with a vanishing error probability and decoding complexity of $\mathcal{O}(u^{3/2}K\log^4 n)$.

References

[1] D. Du, F. K. Hwang, and F. Hwang, *Combinatorial group testing and its applications*, vol. 12. World Scientific, 2000.

[2] H.-B. Chen and F. K. Hwang, "A survey on nonadaptive group testing algorithms through the angle of decoding," *Journal of Combinatorial Optimization*, vol. 15, no. 1, pp. 49–59, 2008.

[3] D. Malioutov and K. Varshney, "Exact rule learning via boolean compressed sensing," in *International Conference on Machine Learning*, pp. 765–773, 2013.

[4] M. T. Goodrich, M. J. Atallah, and R. Tamassia, "Indexing information for data forensics," in *International Conference on Applied Cryptography and Network Security*, pp. 206–221, Springer, 2005.

[5] 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*, 2008.

[6] A. Emad and O. Milenkovic, "Poisson group testing: A probabilistic model for nonadaptive streaming boolean compressed sensing," in *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*, pp. 3335–3339, IEEE, 2014.

[7] K. Chandrasekher, K. Lee, P. Kairouz, R. Pedarsani, and K. Ramchandran, "Asynchronous and noncoherent neighbor discovery for the iot using sparse-graph codes," in *IEEE International Conference on Communications (ICC)*, 2017.

[8] H. Q. Ngo and D.-Z. Du, "A survey on combinatorial group testing algorithms with applications to dna library screening," *Discrete mathematical problems with medical applications*, vol. 55, pp. 171–182, 2000.

[9] S. Cai, M. Jahangoshahi, M. Bakshi, and S. Jaggi, "Grotesque: noisy group testing (quick and efficient)," in *Communication, Control, and Computing (Allerton), 2013 51st Annual Allerton Conference on*, pp. 1234–1241, IEEE, 2013.

[10] K. Lee, R. Pedarsani, and K. Ramchandran, "Saffron: A fast, efficient, and robust framework for group testing based on sparse-graph codes," in *Information Theory (ISIT), 2016 IEEE International Symposium on*, pp. 2873–2877, IEEE, 2016.

[11] P. Damaschke, "Threshold group testing," in *General theory of information transfer and combinatorics*, pp. 707–718, Springer, 2006.

[12] M. Cheraghchi, "Improved constructions for non-adaptive threshold group testing," in *International Colloquium on Automata, Languages, and Programming*, pp. 552–564, Springer, 2010.

[13] G. De Marco, T. Jurdziński, M. Różański, and G. Stachowiak, "Subquadratic non-adaptive threshold group testing," in *International Symposium on Fundamentals of Computation Theory*, pp. 177–189, Springer, 2017.

[14] H.-B. Chen and H.-L. Fu, "Nonadaptive algorithms for threshold group testing," *Discrete Applied Mathematics*, vol. 157, no. 7, pp. 1581–1585, 2009.

[15] T. V. Bui, M. Kuribayashil, M. Cheraghchi, and I. Echizen, "Efficiently decodable non-adaptive threshold group testing," in *2018 IEEE International Symposium on Information Theory (ISIT)*, pp. 2584–2588, IEEE, 2018.

[16] C. L. Chan, S. Cai, M. Bakshi, S. Jaggi, and V. Saligrama, "Stochastic threshold group testing," in *Information Theory Workshop (ITW), 2013 IEEE*, pp. 1–5, IEEE, 2013.