

# Optimal Group Testing using left-and-right-regular sparse-graph codes

Avinash Vem, Nagaraj T. Janakiraman, Krishna R. Narayanan

Department of Electrical and Computer Engineering

Texas A&M University

{vemavinash,tjnagaraj,krn}@tamu.edu

**Abstract**—In this paper, we consider the problem of non-adaptive group testing of  $N$  items out of which  $K$  or less items are known to be defective. We propose a testing scheme based on left-and-right-regular sparse-graph codes with each item being divided into only a finite number of tests  $\ell$ . We show that our scheme requires only  $m = c_1 K \log \frac{c_2 N}{K}$  tests to recover  $(1 - \epsilon)$  fraction of the defective items, for any arbitrarily small  $\epsilon > 0$ , with an asymptotically high probability where the value of constants  $c_1$  and  $\ell$  are a function of the required error floor  $\epsilon$  and the constant  $c_2 \approx 1$  is a function of  $\ell$  and  $c_1$ . More importantly given the output of the testing scheme, the reconstruction algorithm which is iterative and is similar to the peeling decoder, has sub-linear computational complexity of  $O(K \log \frac{N}{K})$  which is known to be optimal. Also for  $m = c_1 K \log K \log \frac{c_2 N}{K}$  tests our scheme recovers all the defective items with asymptotically high probability. These results are valid for both noiseless and noisy versions of the problem as long as the number of defective items scale sub-linearly with the total number of items,  $K = o(N)$ .

## I. INTRODUCTION

The problem of Group Testing (GT) refers to testing a large population for defective items (or sick people) where grouping multiple items together for a single test is possible. The output of the test is *negative* if all the grouped items are non-defective or else *positive*. In the scenario when the fraction of sick people is known to be significantly smaller the objective of GT is to design the grouping scheme such that the total number of tests to be performed is minimized.

This problem was first introduced to the literature of statistics by Dorfman [1] during World War II for testing the soldiers for syphilis without having to test each soldier individually. Since then if we look at the history of the GT problem, depending on the reconstruction guarantees desired, it can be said that there are three different versions of interest: combinatorial, probabilistic and approximate. In the combinatorial designs for the GT problem, the probability of recovering the defective set should be equal to 1 whereas for the probabilistic version one is interested in recovering *all* the defective items with a very high probability either greater than or equal to  $(1 - \epsilon)$  for a given arbitrarily small  $\epsilon > 0$  or approaching 1 asymptotically in  $N$  and  $K$ . For the approximate recovery version one is interested in only recovering a  $(1 - \epsilon)$  fraction of the defective items (not the whole set) with a high probability.

For the combinatorial GT the best known lower bound on the number of tests required is  $\Omega(K^2 \frac{\log N}{\log K})$  [2], [3] whereas the best known achievability bound is  $\Omega(K^2 \log N)$  [4], [5]. Most of these results were based on algorithms relying on exhaustive searches thus have a high computational complexity of at least  $O(K^2 N \log N)$ . Only recently a scheme with efficient decoding was proposed by Indyk et al., [6] where all the defective items are guaranteed to recover using  $m = O(K^2 \log N)$  tests in  $\text{poly}(K) \cdot O(m \log^2 m) + O(m^2)$  time.

If we consider the probabilistic version of the problem, it was shown in [7], [8] that the number of tests necessary is  $\Omega(K \log \frac{N}{K})$  which is the best known lower bound in the literature. And regarding the best known achievability bound Mazumdar [9] proposed a construction that has an asymptotically decaying error probability with  $O(K \frac{\log^2 N}{\log K})$  tests. For the approximate version it was shown [8] that the required number of tests scale as  $O(K \log N)$  and as far as we know this is the tightest bound known.

In [10] authors Lee, Pedarsani and Ramchandran proposed a testing scheme based on *left-regular sparse-graph* codes and a simple *peeling* based decoder, which are popular tools in channel coding [11], for the non-adaptive group testing problem. They refer to the scheme as SAFFRON(Sparse-grAph codes Framewrok For gROup testiNg), a reference which we will follow throughout this paper. The authors proved that using SAFFRON scheme  $m = c_\epsilon K \log_2 N$  number of tests are enough to identify at least  $(1 - \epsilon)$  fraction of defective items (the approximate version of GT) with asymptotically high probability in  $K$  and  $N$ . The precise value of constant  $c_\epsilon$  as a function of the required error floor  $\epsilon$  is also given. More importantly the computational complexity of the proposed peeling based decoder is only  $O(K \log N)$ . They also showed that with  $m = c_\alpha K \log K \log N$  tests i.e. with an additional  $\log K$  factor, the *whole* defective set (the probabilistic version of GT) can be recovered with an asymptotically high probability of  $1 - O(K^{-\alpha})$ .

### A. Our Contributions

In this work, we propose a non-adaptive GT scheme that is similar to the SAFFRON but we employ *left-and-right-regular sparse-graph* codes instead of the left-regular sparse-

graph codes and show that we only require  $c_\epsilon K \log \frac{N}{K}$  number of tests for an error floor of  $\epsilon$  in the approximate version of the GT problem. The other novel result of this construction is that for a fixed error floor  $\epsilon$  we can achieve these optimal testing and sampling complexities using a fixed and finite value for parameter  $\ell$ —the maximum number of tests any item participates in. In other words we are able to solve efficiently for GT (approximate version) under the constraint that the total number of times any item can be divided [12] is finite (finite divisibility constraint). As far as we are aware this is the first scheme which meets the lower bound for approximate GT problem with optimal computational complexity. And also the first testing scheme for the approximate version that achieves the optimal testing complexity under finite divisibility constraint. We also show that for  $m = c \cdot K \log K \log \frac{N}{K}$  tests i.e. with an additional  $\log K$  factor the *whole* defective set can be recovered with an asymptotically high probability of  $1 - O(K^{-\alpha})$ . Note that our testing complexity is only a  $\log K$  factor away from the best known lower bound of  $\Omega(K \log \frac{N}{K})$  [7] for the probabilistic GT problem.

## II. REVIEW: PRIOR WORK

Formally the group testing problem can be stated as following. Given a total number of  $N$  items out of which  $K$  are defective, the objective is to perform  $m$  tests and identify the location of the  $K$  defective items from the test outputs. Each item can participate in multiple tests and the result of a test is positive if and only if atleast one defective item is present.

Let the support vector  $\mathbf{x} \in \{0, 1\}^N$  denote the list of items with the non-zero values corresponding to the defective items. A non-adaptive testing scheme consisting of  $m$  tests can be represented by a matrix  $\mathbf{A} \in \{0, 1\}^{m \times N}$  where each row  $\mathbf{a}_i$  corresponds to a test. The non-zero indices in row  $\mathbf{a}_i$  correspond to the items that participate in  $i^{\text{th}}$  test. The output corresponding to vector  $\mathbf{x}$  and the testing scheme  $\mathbf{A}$  can be expressed in matrix form as:

$$\mathbf{y} = \mathbf{A} \odot \mathbf{x}$$

where  $\odot$  is the usual matrix multiplication in which the arithmetic multiplications are replaced by the boolean AND operation and the arithmetic additions are replaced by the boolean OR operation.

### Testing Scheme

We will briefly review the SAFFRON testing scheme [10], decoding scheme (reconstruction of  $\mathbf{x}$  given  $\mathbf{y}$ ) and their main results in this section. The testing scheme consists of two stages: the first stage is based on a left-regular sparse graph code which pools the  $N$  items into non-disjoint  $M_1$  bins where each item belongs to exactly  $\ell$  bins. The second stage comprises of producing  $h$  testing outputs at each bin where the  $h$  different combinations of the pooled items (from the first stage) at the respective bin are defined according to a universal

signature matrix. For the first stage the authors consider a bipartite graph with  $N$  variable nodes (corresponding to the  $N$  items) and  $M_1$  bin nodes. Each variable node is connected to  $\ell$  bin nodes chosen uniformly at random from the  $M_1$  available bin nodes. All the variable nodes (historically depicted on the left side of the graph in coding theory) have a degree  $\ell$ , hence the left-regular, whereas the degree of a bin node on the right is a random variable ranging from  $[1 : n]$ .

**Definition 1** (Left-regular sparse graph ensemble). We define  $\mathcal{G}_\ell(N, M_1)$  to be the ensemble of left-regular bipartite graphs where for each variable node the  $\ell$  right node connections are chosen uniformly at random from the  $M_1$  right nodes.

Let  $\mathbf{T}_G \in \{0, 1\}^{M_1 \times N}$  be the adjacency matrix corresponding to a graph  $G \in \mathcal{G}_\ell(N, M_1)$  i.e., each column in  $\mathbf{T}_G$  corresponds to a variable node and has exactly  $\ell$  ones. Let the rows in matrix  $\mathbf{T}_G$  be given by  $\mathbf{T}_G = [\mathbf{t}_1^T, \mathbf{t}_2^T, \dots, \mathbf{t}_{M_1}^T]^T$ . For the second stage let the universal signature matrix defining the  $h$  tests at each bin be  $\mathbf{U} \in \{0, 1\}^{h \times N}$ . Then the overall testing matrix  $\mathbf{A} := [\mathbf{A}_1^T, \dots, \mathbf{A}_{M_1}^T]^T$  where  $\mathbf{A}_i = \mathbf{U} \text{diag}(\mathbf{t}_i)$  of size  $h \times N$  defines the  $h$  tests at  $i^{\text{th}}$  bin. Thus the total number of tests is  $M = M_1 \times h$ .

The signature matrix  $\mathbf{U}$  in a more general setting with parameters  $r$  and  $p$  can be given by

$$\mathbf{U}_{r,p} = \begin{bmatrix} \mathbf{b}_1 & \mathbf{b}_2 & \cdots & \mathbf{b}_r \\ \bar{\mathbf{b}}_1 & \bar{\mathbf{b}}_2 & \cdots & \bar{\mathbf{b}}_r \\ \mathbf{b}_{\pi_1^1} & \mathbf{b}_{\pi_2^1} & \cdots & \mathbf{b}_{\pi_r^1} \\ \bar{\mathbf{b}}_{\pi_1^1} & \bar{\mathbf{b}}_{\pi_2^1} & \cdots & \bar{\mathbf{b}}_{\pi_r^1} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{b}_{\pi_1^p} & \mathbf{b}_{\pi_2^p} & \cdots & \mathbf{b}_{\pi_r^p} \\ \bar{\mathbf{b}}_{\pi_1^p} & \bar{\mathbf{b}}_{\pi_2^p} & \cdots & \bar{\mathbf{b}}_{\pi_r^p} \end{bmatrix} \quad (1)$$

where  $\mathbf{b}_i \in \{0, 1\}^{\lceil \log_2 r \rceil}$  is the binary expansion vector for  $i$  and  $\bar{\mathbf{b}}_i$  is the complement of  $\mathbf{b}_i$ .  $\pi^k = [\pi_1^k, \pi_2^k, \dots, \pi_r^k]$  denotes a permutation chosen at random from symmetric group  $S_r$ . Henceforth  $\mathbf{U}_{r,p}$  will refer to either the ensemble of matrices generated over the choices of the permutations  $\pi^k$  for  $k \in [1 : p]$  or a matrix picked uniformly at random from the said ensemble. The reference should be sufficiently clear from the context. In the SAFFRON scheme the authors employ a signature matrix that is equivalent to  $\mathbf{U}_{r,p}$  with  $r = N$  and  $p = 2$  thus resulting in a  $\mathbf{U}$  of size  $h \times N$  with  $h = 6 \log_2 N$ .

### Decoding

Before describing the decoding process let us review some terminology. A bin node is referred to as a *singleton* if there is exactly one non-zero variable node connected and similarly referred to as a *double-ton* in case of two non-zero variable nodes. In the case where we know the identity of one of them leaving the decoder to decode the identity of the other one, the bin is referred to as a *resolvable double-ton*. And if the bin has more than two non-zero variable nodes attached we refer to it as a *multi-ton*.

First part of the decoder which is referred to as bin decoder will be able to detect and decode exactly the identity of the non-zero variable nodes connected to the bin if and only if the bin is a singleton or a resolvable double-ton. If the bin is a multi-ton the bin decoder will detect it neither as a singleton or a resolvable double-ton with high probability. The second part of the decoder which is commonly referred to as peeling decoder [13], when given the identities of some of the non-zero variable nodes by the bin decoder, identifies the bins connected to the recovered variable nodes and looks for newly uncovered resolvable double-ton in these bins. This process of recovering new non-zero variable nodes from already discovered non-zero variable nodes proceeds in an iterative manner (referred to as peeling off from the graph historically). For details of the decoder we refer the reader to [10].

**Remark 2.** Note that we are not literally peeling off the decoded nodes from the graph because of the *non-linear* OR operation on the non-zero variable nodes at each bin thus preventing us in subtracting the effect of the non-zero node from the measurements of the bin node unlike in the problems of compressed sensing or LDPC codes on binary erasure channel.

Now we state the series of lemmas and theorems, without proofs, from [10] that shows that the SAFFRON scheme with the described peeling decoder solves the approximate GT with  $\Omega(K \log N)$  tests and  $O(K \log N)$  computational complexity.

**Lemma 3** (Bin decoder analysis). For a signature matrix  $\mathbf{U}_{r,p}$  as described in (1), the bin decoder successfully detects and resolves if the bin is either a singleton or a resolvable double-ton. In the case of the bin being a multi-ton, the bin decoder declares a wrong hypothesis of either a singleton or a resolvable double-ton with a probability no greater than  $\frac{1}{r^b}$ .

For convenience the performance of the peeling decoder is analyzed independently of the bin decoder i.e., a peeling decoder is considered which assumes that the bin decoder is working accurately which will be referred to as *oracle based peeling decoder*. Another simplification is that a pruned graph is considered where all the zero variable nodes and their respective edges are removed from the graph. Also the oracle based peeling decoder is assumed to decode a variable node if it is connected to a bin-node with degree one or degree two with one of them already decoded, in an iterative fashion. Any right node with more than degree two is untouched by this oracle based peeling decoder. It is easy to verify that the original decoder with accurate bin decoding is equivalent to this simplified oracle based peeling decoder on a pruned graph.

**Definition 4** (Pruned graph ensemble). We will define the pruned graph ensemble  $\tilde{\mathcal{G}}_\ell(N, K, M_1)$  as the set of all bipartite graphs obtained from removing a random  $N - K$  subset of variable nodes from a graph from the ensemble  $\mathcal{G}_\ell(N, M_1)$ . Note that graphs from the pruned ensemble have  $K$  variable nodes.

$\epsilon$	$10^{-3}$	$10^{-4}$	$10^{-5}$	$10^{-6}$	$10^{-7}$	$10^{-8}$	$10^{-9}$
$c_\epsilon$	6.13	7.88	9.63	11.36	13.10	14.84	16.57
$\ell$	7	9	10	12	14	15	17

TABLE I  
CONSTANTS FOR VARIOUS ERROR FLOOR VALUES

Before we analyze the pruned graph ensemble let us define the right-node degree distribution (d.d) of an ensemble as  $R(x) = \sum_i R_i x^i$  where  $R_i$  is the probability that a right-node has degree  $i$ . Similarly the edge d.d  $\rho(x) = \sum_i \rho_i x^{i-1}$  is defined where  $\rho_i$  is the probability that a random edge in the graph is connected to a right-node of degree  $i$ . Note that the left-degree distribution is regular even for the pruned graph ensemble and hence is not specifically mentioned.

**Lemma 5** (Edge d.d of Pruned graph). For the pruned ensemble  $\tilde{\mathcal{G}}_\ell(N, K, M_1)$ , it can be shown in the limit  $K, N \rightarrow \infty$  that  $\rho_1 = e^{-\lambda}$  and  $\rho_2 = \lambda e^{-\lambda}$  where  $\lambda = l/c_1$  if  $M_1 = c_1 K$  for some constant  $c_1$ .

**Lemma 6.** For the pruned graph ensemble  $\tilde{\mathcal{G}}_\ell(N, K, M_1)$  the oracle-based peeling decoder fails to decode atleast  $(1 - \epsilon)$  fraction of the variable nodes with exponentially decaying probability if  $M_1 \geq c_\epsilon K$  where  $c_\epsilon$  for various  $\epsilon$  is given in Table. I.

*Proof.* Instead of reworking the whole proof here from [10], we will list the main steps involved in the proof which we will use further along. If we let  $p_j$  be the probability that a random defective item is not identified at iteration  $j$ , in the limit  $N$  and  $K \rightarrow \infty$  we can write density evolution (DE) equation relating  $p_{j+1}$  to  $p_j$  as

$$p_{j+1} = [1 - (\rho_1 + \rho_2(1 - p_j))]^{\ell-1}.$$

For this DE, we can see that 0 is not a fixed point and hence  $p_j \rightarrow 0$  as  $j \rightarrow \infty$ . Therefore numerically optimizing the values of  $c_\epsilon$  and  $\ell$  such that  $\lim_{j \rightarrow \infty} p_j \leq \epsilon$  gives the optimal values for  $c_\epsilon$  and  $\ell$  given in Table. I.  $\square$

Combining the lemmas and remarks above, the main result from [10] can be summarized as follows.

**Theorem 7.** For any arbitrarily-small  $\epsilon > 0$  the SAFFRON framework performing  $m = 6c_\epsilon K \log_2 N$  tests recovers atleast  $(1 - \epsilon)$  fraction of the defective items with a high probability of atleast  $1 - O(\frac{K}{N^2})$ . And the computational complexity of the decoding scheme is  $O(K \log N)$ . The constant  $c_\epsilon$  is given in Table. I for some values of  $\epsilon$ .

### III. PROPOSED SCHEME

The main difference between the SAFFRON scheme and our proposed scheme is that we use a left-and-right-regular sparse-graph in the first stage for the binning operation.

**Definition 8** (Left-and-right-regular sparse graph ensemble). We define  $\mathcal{G}_{\ell,r}(N, M_1)$  to be the ensemble of left-and-right-regular graphs where the  $N\ell$  edge connections from the left

and  $M_1 r (= N\ell)$  edge connections from the right are paired up according to a permutation  $\pi_{N\ell}$  chosen at random.

Let  $\mathbf{T}_G \in \{0,1\}^{M_1 \times N}$  be the adjacency matrix corresponding to a graph  $G \in \mathcal{G}_{\ell,r}(N, M_1)$  i.e., each column in  $\mathbf{T}_G$  corresponding to a variable node has exactly  $\ell$  ones and each row corresponding to a bin node has exactly  $r$  ones. And let the universal signature matrix be  $\mathbf{U} \in \{0,1\}^{h \times r}$  chosen from the  $\mathbf{U}_{r,p}$  ensemble. Then the overall testing matrix  $\mathbf{A} := [\mathbf{A}_1^T, \dots, \mathbf{A}_{M_1}^T]^T$  where  $\mathbf{A}_i \in \{0,1\}^{h \times N}$  defining the  $h$  tests at  $i^{\text{th}}$  bin is given by

$$\mathbf{A}_i = [\mathbf{0}, \dots, \mathbf{0}, \mathbf{u}_1, \mathbf{0}, \dots, \mathbf{u}_2, \mathbf{0}, \dots, \mathbf{u}_r], \quad \text{where} \quad (2)$$

$$\mathbf{t}_i = [0, \dots, 0, 1, 0, \dots, 1, 0, \dots, 1].$$

Note that  $\mathbf{A}_i$  is defined by placing the  $r$  columns of  $\mathbf{U}$  at the  $r$  non-zero indices of  $\mathbf{t}_i$  and the remaining are padded with zero columns. We can observe that the total number of tests for this scheme is  $M = M_1 \times h$  where  $h = (2p+2) \log r$ .

**Definition 9** (Regular SAFFRON). We define the ensemble of testing matrices for our scheme to be  $\mathcal{G}_{\ell,r}(N, M_1) \times \mathbf{U}_{r,p}$  where a graph  $G$  is chosen from  $\mathcal{G}_{\ell,r}(N, M_1)$ , a signature matrix  $\mathbf{U}$  is chosen from  $\mathbf{U}_{r,p}$  and the testing matrix is defined according to Eq. (2). Note that the total number of tests for this testing scheme is  $(2p+2)M_1 \log r$  where  $r = \frac{N\ell}{M_1}$ .

For the regular SAFFRON testing ensemble defined in Def. 9, we employ the same peeling based decoder described in Sec. II. Similar to the SAFFRON scheme we will analyze the peeling decoder and the bin decoder separately and union bound the total error probability of the decoding scheme. As we have already mentioned the analysis of the peeling decoder part alone can be carried out by considering a *simplified oracle-based peeling decoder* on a pruned graph with only the non-zero variable nodes remaining.

**Definition 10** (Pruned graph ensemble). We will define the pruned graph ensemble  $\tilde{\mathcal{G}}_{\ell,r}(N, K, M_1)$  as the set of all graphs obtained from removing a random  $N - K$  subset of variable nodes from a graph from the ensemble  $\mathcal{G}_{\ell,r}(N, M_1)$ .

Note that graphs from the pruned ensemble have  $K$  variable nodes with a degree  $\ell$  whereas the right degree is not regular anymore.

**Lemma 11** (Edge d.d of pruned graph). For the pruned graph ensemble  $\tilde{\mathcal{G}}_{\ell,r}(N, K, M_1)$  it can be shown in the limit  $K, N \rightarrow \infty$  that edge d.d coefficients are  $\rho_1 = e^{-\lambda}$  and  $\rho_2 = \lambda e^{-\lambda}$  where  $\lambda = \ell/c_\epsilon$  for the choice of  $M_1 = c_\epsilon K$ ,  $c_\epsilon$  being some constant.

Note that even if our initial ensemble is left-and-right-regular the pruned graph has asymptotically same degree distribution as in the SAFFRON scheme where the initial graph is from left-regular ensemble.

**Lemma 12.** For the pruned graph ensemble  $\tilde{\mathcal{G}}_{\ell,r}(N, K, M_1)$  the oracle-based peeling decoder fails to peel off at least  $(1 -$

$\epsilon)$  fraction of the variable nodes with exponentially decaying probability for  $M_1 = c_\epsilon K$  where  $c_\epsilon$  for various  $\epsilon$  is given in Table. I.

*Proof.* From Lemma. 11 we know that the edge degree distribution coefficients  $\rho_1$  and  $\rho_2$  are identical to that of the SAFFRON scheme and hence the same DE equations can be used here. Therefore the exact same proof as the proof of Lemma. 6 can be employed here.  $\square$

**Theorem 13.** Let  $p \in \mathbb{Z}$  such that  $K$  and  $N$  scale as  $K \in o(N^{\frac{p}{p+1}})$ . The proposed regular SAFFRON framework using  $M = 2(p+1)c_\epsilon K \log_2 \frac{N\ell}{c_\epsilon K}$  tests recovers atleast  $(1 - \epsilon)$  fraction of the defective items with asymptotically high probability of  $1 - O\left(\frac{K^{p+1}}{N^p}\right)$ , for any arbitrarily small  $\epsilon > 0$ . Note that computational complexity of the decoding scheme is  $O(K \log \frac{N}{K})$ . The constant  $c_\epsilon$  is given in Table. I.

*Proof.* It remains to be shown that the total probability of error decays asymptotically in  $K$  and  $N$ . Let  $E_1$  be the event of oracle-based peeling decoder terminating without recovering atleast  $(1 - \epsilon)K$  variable nodes. Let  $E_2$  be the event of the bin decoder making an error during the entirety of the peeling process and  $E_{\text{bin}}$  be the event of one instance of bin decoder making an error. The total probability of error  $P_e$  can be upper bounded by

$$\begin{aligned} P_e &\leq \Pr(E_1) + \Pr(E_2) \\ &\leq \Pr(E_1) + K\ell \Pr(E_{\text{bin}}) \\ &\in O\left(\frac{K^{p+1}}{N^p}\right) \end{aligned}$$

where the second inequality is due to the union bound over a maximum of  $K\ell$  instances of bin decoding. The third line is due to the fact that  $\Pr(E_1)$  is exponentially decaying in  $K$  (see Lemma. 12) and  $\Pr(E_{\text{bin}}) = (\frac{c_1 K}{N\ell})^p$  (see Lemma. 3 and Def. 9)  $\square$

*Proof of Lem. 11.* We will first derive  $R(x)$  for the pruned graph ensemble and then use the relation [11]  $\rho(x) = \frac{R'(x)}{R'(1)}$  to derive the edge d.d. Note that all the check nodes have a uniform degree  $r$  before pruning. When pruning we are removing a  $N - K$  subset of variable nodes at random i.e., asymptotically this is equivalent to removing each edge from the graph with a probability  $1 - \beta$  where  $\beta := \frac{K}{N}$ . Under this process the right-node d.d can be written as

$$\begin{aligned} R_1 &= r\beta(1 - \beta)^{r-1}, \quad \text{and similarly} \quad (3) \\ R_i &= \binom{r}{i} \beta^i (1 - \beta)^{r-i}, \end{aligned}$$

thus giving us  $R(x) = (\beta x + (1 - \beta))^r$ . This gives us

$$\begin{aligned} \rho(x) &= \frac{\beta(\epsilon x + (1 - \beta))^{r-1}}{r\beta} \\ &= (\beta x + (1 - \beta))^{r-1}. \end{aligned}$$

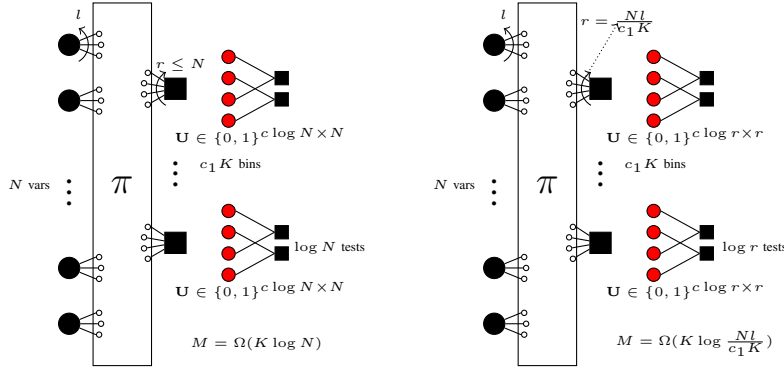


Fig. 1. Illustration of the main differences between SAFFRON [10] on the left and our regular-SAFFRON scheme on the right. In both the schemes the peeling decoder on sparse graph requires  $\Omega(K)$  bins. But for the bin decoder part, in SAFFRON scheme the right degree is a random variable with a maximum value of  $N$  and thus requires  $\Omega(\log N)$  tests at each bin. Whereas our scheme based on right-regular sparse graph has a constant right degree of  $\Omega(\frac{N}{K})$  and thus requires only  $\Omega(\log \frac{N}{K})$  tests at each bin. Thus we can improve the number of tests from  $\Omega(K \log N)$  to order optimal  $\Omega(K \log \frac{N}{K})$ .

Thus we can compute that  $\rho_1 = (1 - \beta)^{r-1}$  and  $\rho_2 = (r - 1)\beta(1 - \beta)^{r-2}$ . For  $M_1 = c_\epsilon K$  we evaluate these quantities in the limit  $K, N \rightarrow \infty$  as

$$\begin{aligned} \lim_{K, N \rightarrow \infty} \rho_1 &= \lim_{K, N \rightarrow \infty} \left(1 - \frac{K}{N}\right)^{\frac{N\ell}{c_\epsilon K} - 1} \\ &= e^{-\lambda} \quad \text{where } \lambda = \frac{\ell}{c_\epsilon} \end{aligned}$$

Similarly we can show  $\lim_{K, N \rightarrow \infty} \rho_2 = \lambda e^{-\lambda}$ .  $\square$

#### IV. TOTAL RECOVERY: SINGLETON-ONLY VARIANT

In this section we will look at the proposed regular-SAFFRON scheme but with a decoder that uses only the singleton bins. To elaborate the decoder is not iterative and recovers all the variable nodes connected to the degree-1 check nodes and terminates. We will refer to this scheme as *singleton-only* regular-SAFFRON scheme. The downside of this using this scheme, even though we can now recover the *whole* defective set instead of just a large fraction of the defective items, is that now the left-degree of the bipartite graph, or equivalently the number of times a sample is divided for testing, cannot be finite and needs to be scaled as  $\log K$ .

Since we do not need to be able to recover resolvable double-tons we only need  $2 \log_2 r$  number of tests at each bin i.e.  $p = 0$  in Eqn. (1).

**Theorem 14.** Using a  $(\ell, r) = (c_\alpha \log K, \frac{N}{K})$  regular-bipartite graph, the regular-SAFFRON scheme with  $m = 2c_\alpha K \log K \log_2 \frac{N}{K}$  tests and the singleton-only decoder fails to recover all the non-zero variable nodes with a vanishing probability of  $O(\frac{1}{K^\alpha})$  where  $c_\alpha = e(1 + \alpha)$ .

*Proof.* As the number of tests in each bin is  $2 \log_2 \frac{N}{K}$  it is enough if we show that for the number of bins in the bipartite graph equal to  $e(1 + \alpha)K \log K$  all the variable nodes in the pruned graph are connected to atleast one singleton bin with high probability.

In the pruned graph ensemble, for any particular variable node, the probability that any of the  $\ell$  connected bit nodes are not a singleton can be given by  $(1 - R_1)^\ell$  where  $R_1$  is the probability that a bin node in the pruned graph ensemble is a singleton. From Eq. 3 asymptotically the value of  $R_1$  approaches

$$\begin{aligned} R_1 &= \lim_{K, N \rightarrow \infty} r\beta(1 - \beta)^{r-1} \\ &= \lim_{K, N \rightarrow \infty} \left(1 - \frac{K}{N}\right)^{\frac{N}{K} - 1} \\ &\approx e^{-1} \end{aligned}$$

By using union bound over all the  $K$  variable nodes in the pruned graph, the probability  $P_e$  that the singleton-only decoder fails to recover a defective item can be bounded by

$$\begin{aligned} P_e &\leq K(1 - R_1)^\ell \\ &\approx K(1 - e^{-1})^{e(1 + \alpha) \log K} \\ &\leq K e^{-e^{-1} e(1 + \alpha) \log K} \\ &= K^{-\alpha}, \end{aligned}$$

where for the third line we employ  $(1 - x) \leq e^{-x}$   $\square$

#### V. ROBUST GROUP TESTING

In this section we extend our scheme to the group testing problem where the test results can be noisy,

$$\mathbf{y} = \mathbf{A} \odot \mathbf{x} + \mathbf{w},$$

where the addition is over binary field and  $\mathbf{w} \in \{0, 1\}^N$  is an i.i.d noise vector distributed according to Bernoulli distribution with parameter  $0 < q < \frac{1}{2}$ .

##### Testing Scheme

Let  $\mathcal{C}_n$  be a binary error-correcting code with the following definition:

- Let the encoder and decoder functions be  $f : \{0, 1\}^n \rightarrow \{0, 1\}^{\frac{n}{R}}$  and  $g : \{0, 1\}^{\frac{n}{R}} \rightarrow \{0, 1\}^n$  respectively where  $R$  is the rate of the code.

For the choice of  $\mathcal{C}_n$  we will use spatially-coupled LDPC codes which are known to be capacity achieving [14], [15]. A sequence of codes being capacity achieving is equivalent to:

- There exists a sequence of codes  $\{\mathcal{C}_n\}$  with the rate of each code being  $R$  satisfying

$$R < 1 - H(q) - \delta = 1 + q \log_2 q + \bar{q} \log_2 \bar{q} - \delta \quad (4)$$

for any arbitrary small constant  $\delta$  such that the probability of error  $\Pr(g(\mathbf{x} + \mathbf{w}) \neq \mathbf{x}) < 2^{-\kappa n}$  for some  $\kappa > 0$ . In Eqn. 4,  $\bar{q} := 1 - q$ .

The modified signature matrix  $\mathbf{U}'_{r,p}$  can be described via  $\mathbf{U}_{r,p}$  given in Eq. (1) and encoding function  $f$  for  $\mathcal{C}_n$  where  $n = \lceil \log_2 r \rceil$  as follows:

$$\mathbf{U}'_{r,p} := \begin{bmatrix} f(\mathbf{b}_1) & f(\mathbf{b}_2) & \cdots & f(\mathbf{b}_r) \\ f(\mathbf{b}_1) & f(\mathbf{b}_2) & \cdots & f(\mathbf{b}_r) \\ f(\mathbf{b}_{\pi_1}) & f(\mathbf{b}_{\pi_2}) & \cdots & f(\mathbf{b}_{\pi_r}) \\ f(\mathbf{b}_{\pi_1}) & f(\mathbf{b}_{\pi_2}) & \cdots & f(\mathbf{b}_{\pi_r}) \\ \vdots & \vdots & \ddots & \vdots \\ f(\mathbf{b}_{\pi_1^p}) & f(\mathbf{b}_{\pi_2^p}) & \cdots & f(\mathbf{b}_{\pi_r^p}) \\ f(\mathbf{b}_{\pi_1^p}) & f(\mathbf{b}_{\pi_2^p}) & \cdots & f(\mathbf{b}_{\pi_r^p}) \end{bmatrix} \quad (5)$$

Then the overall testing matrix  $\mathbf{A}$  is defined similar to the noiseless case in Sec. II except that  $\mathbf{U}$  will be replaced by  $\mathbf{U}'$  in Eqn. (5).

### Decoding

The iterative part of the decoder is identical to that of the noiseless case whereas the bin detection part differs slightly. Given the test output vector at a bin  $\mathbf{y} = [\mathbf{y}_{01}^T, \mathbf{y}_{02}^T, \mathbf{y}_{11}^T, \dots, \mathbf{y}_{p2}^T]^T$ , the bin decoder first applies the decoding function  $g(\cdot)$  to the first segments in each section  $\mathbf{y}_{i1} \forall i \in [0 : p]$  then obtains the locations  $l_0, l_1, \dots, l_p$  whose binary expansions are equal to the channel decoder outputs  $g(\mathbf{y}_{i1})$ . A singleton is declared if  $\pi_{l_0}^i = l_i \forall i$ .

Consider the resolvable double-ton decoding. Let the location of the already recovered variable node in the double-ton bin be  $l_0$ . Then the test output can be given as

$$\begin{bmatrix} \mathbf{y}_{01} \\ \mathbf{y}_{02} \\ \mathbf{y}_{11} \\ \vdots \\ \mathbf{y}_{p2} \end{bmatrix} = \mathbf{u}_{l_0} \vee \mathbf{u}_{l_1} + \mathbf{w} = \begin{bmatrix} f(\mathbf{b}_{l_0}) \\ f(\mathbf{b}_{l_0}) \\ f(\mathbf{b}_{\pi_{l_0}^1}) \\ \vdots \\ f(\mathbf{b}_{\pi_{l_0}^p}) \end{bmatrix} \vee \begin{bmatrix} f(\mathbf{b}_{l_1}) \\ f(\mathbf{b}_{l_1}) \\ f(\mathbf{b}_{\pi_{l_1}^1}) \\ \vdots \\ f(\mathbf{b}_{\pi_{l_1}^p}) \end{bmatrix} + \begin{bmatrix} \mathbf{w}_{01} \\ \mathbf{w}_{02} \\ \mathbf{w}_{11} \\ \vdots \\ \mathbf{w}_{p2} \end{bmatrix}$$

where the location of the second non-zero variable node  $l_1$  needs to be recovered. Given  $\mathbf{y} = \mathbf{u}_{l_0} \vee \mathbf{u}_{l_1} + \mathbf{w}$  and  $\mathbf{u}_{l_0}$ ,  $f(\mathbf{b}_{\pi_{l_1}^k}) + \mathbf{w}$  can be recovered since for each segment either the vector  $f(\mathbf{b}_{\pi_{l_1}^k})$  or its complement is available. Once  $f(\mathbf{b}_{\pi_{l_1}^k}) +$

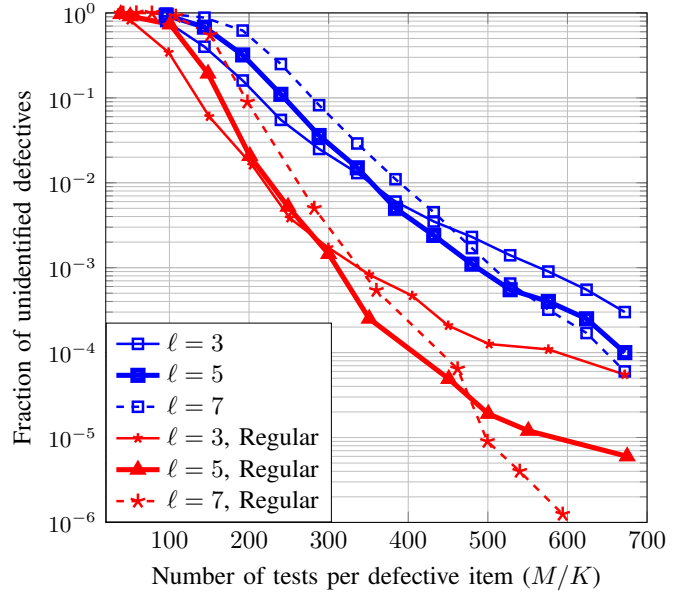


Fig. 2. MonteCarlo simulations for  $K = 100, N = 2^{16}$ . We compare the SAFFRON scheme with our regular SAFFRON scheme for various left degrees  $\ell \in \{3, 5, 7\}$ . For a given  $\ell$  the bin detection size is fixed and we vary the number of bins. The plots in blue indicate the SAFFRON scheme and the plots in red indicate our regular SAFFRON scheme based on left-and-right-regular bipartite graphs.

$\mathbf{w}, \forall k$  are recovered, we apply singleton decoding procedure and rules as described above.

**Lemma 15** (Robust Bin Decoder Analysis). For  $\mathbf{U}'_{r,p}$  in (5), the robust bin decoder misses a singleton with probability no greater than  $\frac{p+1}{r^\kappa}$ . The robust bin decoder wrongly declares a singleton with probability at most  $\frac{1}{r^{p+p\kappa}}$ .

The fraction of missed singletons can be compensated by using  $M(1 + \frac{p+1}{r^\kappa})$  instead of  $M$  such that the total number of singletons decoded will be  $M(1 + \frac{p+1}{r^\kappa})(1 - \frac{p+1}{r^\kappa}) \approx M$ .

**Theorem 16.** Let  $p \in \mathbb{Z}$  such that  $K$  and  $N$  scale as  $K \in o(N^{(p-1)/p})$ . The proposed robust regular SAFFRON framework using  $M = 2(p+1)\beta(q)c_\epsilon K \log_2 \frac{N}{K}$  tests recovers atleast  $(1 - \epsilon)$  fraction of the defective items, for any arbitrarily small  $\epsilon > 0$ , with asymptotically high probability of  $1 - O(N^{-\kappa})$ . Here  $\beta(q) = \frac{1}{R} > \frac{1}{1-H(q)-\delta}$  for some  $\delta > 0$  and the constant  $c_\epsilon$  is given in Table. I. The computational complexity of the decoding scheme is  $O(K \log \frac{N}{K})$ .

*Proof.* Similar to the noiseless case the total probability of error  $P_e$  is dominated by the performance of bin decoder.

$$\begin{aligned} P_e &\leq \Pr(E_1) + K\ell\Pr(E_{\text{bin}}) \\ &\leq \Pr(E_1) + \frac{K^{1+p(1+\kappa)}}{N^{p(1+\kappa)}} \\ &O(N^{-\kappa}) \end{aligned}$$

where the last line is due to the fact that  $\Pr(E_1)$  is exponentially decaying in  $K$  and  $K \leq N^{(p-1)/p}$  for large enough

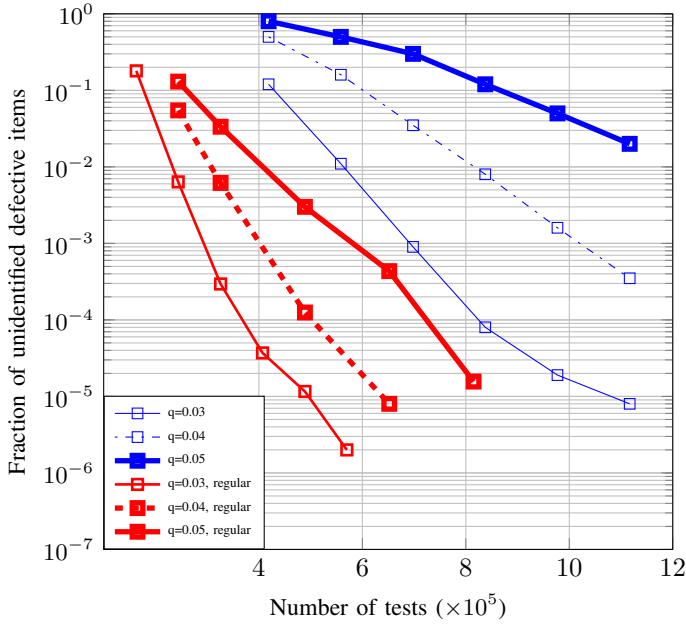


Fig. 3. Monte Carlo simulations for  $K = 128, N = 2^{32}$ . We compare the SAFFRON scheme with our regular SAFFRON scheme for a left degree  $\ell = 12$ . We fix the number of bins and vary the rate of the error control code used. The plots in blue indicate the SAFFRON scheme [10] and the plots in red indicate our regular SAFFRON scheme based on left-and-right-regular bipartite graphs.

$K, N$ .

□

## VI. SIMULATION RESULTS

In this section we will evaluate the performance of our proposed regular SAFFRON scheme via Monte Carlo simulations and compare it with the results for SAFFRON scheme provided in [10].

### Noiseless Group Testing

We demonstrate the result in Thm. 13 by simulating the performance for the system parameters summarized below.

- We fix  $N = 2^{16}$  and  $K = 100$
- In Eqn. 1 the parameter  $p = 2$  is chosen i.e.  $h = 6 \log_2 r$
- For each  $\ell \in \{3, 5, 7\}$  we vary the number of bins  $M_1 = cK$ . Value of  $r = \frac{N\ell}{cK}$  changes accordingly.
- Hence the total number of tests  $M = 6cK \log_2 \left( \frac{N\ell}{cK} \right)$

The results are shown in Fig. 2. We observe that there is clear improvement in performance for our regular SAFFRON scheme when compared to the SAFFRON scheme for each  $\ell \in \{3, 5, 7\}$ .

### Noisy Group Testing

Similar to the noiseless group testing problem we simulate the performance of our robust regular SAFFRON scheme

and compare it with that of the SAFFRON scheme. For convenience of comparison we choose our system parameters identical to the choices in [10]. The system parameters can be summarized as follows:

- $N = 2^{32}, K = 2^7$ . We fix  $\ell = 12, M_1 = 11.36K$
- BSC noise parameter  $q \in \{0.03, 0.04, 0.05\}$

For the signature matrix we choose  $p = 1$  in Eqn. 1 i.e.  $h = 4 \log_2 r$ . Note that for the above set of parameters the right degree  $r = \frac{N\ell}{M_1} \approx 26$ . We choose to operate in field  $GF(2^7)$  thus giving us a message length of 4 symbols. Then we encode using a  $(4 + 2e, 4)$  Reed-Solomon code for  $e \in [0 : 8]$  thus giving us a column length of  $4 \times 7(4 + 2e)$  bits and the total number of tests  $M = 28M_1(4 + 2e)$ .

## REFERENCES

- [1] R. Dorfman, "The detection of defective members of large populations," *The Annals of Mathematical Statistics*, vol. 14, no. 4, pp. 436–440, 1943.
- [2] A. G. D'yachkov and V. V. Rykov, "Bounds on the length of disjunctive codes," *Problemy Peredachi Informatsii*, vol. 18, no. 3, pp. 7–13, 1982.
- [3] P. Erdős, P. Frankl, and Z. Füredi, "Families of finite sets in which no set is covered by the union of others," *Israel Journal of Mathematics*, vol. 51, no. 1, pp. 79–89, 1985.
- [4] W. Kautz and R. Singleton, "Nonrandom binary superimposed codes," *IEEE Transactions on Information Theory*, vol. 10, no. 4, pp. 363–377, 1964.
- [5] E. Porat and A. Rothschild, "Explicit nonadaptive combinatorial group testing schemes," *IEEE Transactions on Information Theory*, vol. 57, no. 12, pp. 7982–7989, 2011.
- [6] P. Indyk, H. Q. Ngo, and A. Rudra, "Efficiently decodable non-adaptive group testing," in *Proceedings of the twenty-first annual ACM-SIAM symposium on Discrete Algorithms*, pp. 1126–1142, Society for Industrial and Applied Mathematics, 2010.
- [7] C. L. Chan, S. Jaggi, V. Saligrama, and S. Agnihotri, "Non-adaptive group testing: Explicit bounds and novel algorithms," *IEEE Trans. Inf. Theory*, vol. 60, no. 5, pp. 3019–3035, 2014.
- [8] G. K. Atia and V. Saligrama, "Boolean compressed sensing and noisy group testing," *IEEE Trans. Inf. Theory*, vol. 58, no. 3, pp. 1880–1901, 2012.
- [9] A. Mazumdar, "Nonadaptive group testing with random set of defectives via constant-weight codes," *arXiv preprint arXiv:1503.03597*, 2015.
- [10] K. Lee, R. Pedarsani, and K. Ramchandran, "Saffron: A fast, efficient, and robust framework for group testing based on sparse-graph codes," *arXiv preprint arXiv:1508.04485*, 2015.
- [11] T. Richardson and R. Urbanke, *Modern coding theory*. Cambridge University Press, 2008.
- [12] V. Gandikota, E. Grigorescu, S. Jaggi, and S. Zhou, "Nearly optimal sparse group testing."
- [13] X. Li, S. Pawar, and K. Ramchandran, "Sub-linear time compressed sensing using sparse-graph codes," in *Proc. IEEE Int. Symp. Information Theory*, pp. 1645–1649, 2015.
- [14] S. Kumar, A. J. Young, N. Macris, and H. D. Pfister, "Threshold saturation for spatially coupled ldpc and ldgm codes on bms channels," *IEEE Trans. Inf. Theory*, vol. 60, no. 12, pp. 7389–7415, 2014.
- [15] S. Kudekar, T. Richardson, and R. L. Urbanke, "Spatially coupled ensembles universally achieve capacity under belief propagation," *IEEE Trans. Inf. Theory*, vol. 59, no. 12, pp. 7761–7813, 2013.