Algorithms for Locating Non-Defective Items in a Large Population

THESIS

Submitted in partial fulfillment of the requirements of BITS C421T/422T Thesis

By

KULKARNI ANUVA ABHIJIT

2009A3TS005G

Under the supervision of

Dr. Chandra R. Murthy

Signal Processing for Communications Lab

Dept. of ECE

INDIAN INSTITUTE OF SCIENCE, BANGALORE

BITS, PILANI - K.K.BIRLA GOA CAMPUS

5 May 2013

Thesis Title : Algorithms for Locating Non-Defective Items in a Large Population

Supervisor : Dr. Chandra R. Murthy

Semester : Second **Year** : 2012 - 2013

Name of the Student : Kulkarni Anuva Abhijit **ID. No**. : 2009A3TS005G

Thesis Abstract :

Combinatorial group testing provides an efficient way to identify the defective items in a given population, provided the size of defective set is small compared to the population size.

The basic idea in group testing is to identify a small number of defective items from a large collection of items, while minimizing the number of tests required to do so. Some applications where group testing is used are : DNA sequencing, testing chemicals, drugs or vaccines for contamination, detection of faulty nodes or links in wireless networks, etc.

Many times, however, it is more appropriate to identify a subset of "non-defectives" instead of all the "defectives". For instance, in a cognitive radio network, a secondary network needs to find a certain amount of free spectrum for setting up secondary operations. In the group testing terminology, the frequency sub-bands used by the primary users are the defective items and the sub-bands not used by the primary users are the non-defective items.

Note that the number of free sub-bands (non-defective items) required by the secondary network can be much less than the available number of frequency sub-bands. Thus, in this application the main goal is to find an unused chunk of spectrum for the use by the secondary network, i.e., identification of a subset of non-defective items. The methods used in group testing are an indirect way to find the non-defective items, since they focus on identifying the defective items first, thereby identifying all the non-defective items also.

The focus of our project is to find methods for identifying a subset of the non-defective items in a non-adaptive group testing setup. In this thesis, we investigate certain algorithms for non-adaptive group testing for noisy and noiseless cases. For our main problem of directly identifying a subset of non-defectives, we emphasize on finding methods which are computationally tractable and have provable recovery performance guarantees.

CERTIFICATE

This is to certify that the thesis entitled **Algorithms for Locating Non-Defective Items in a Large Population** submitted by **Kulkarni Anuva Abhijit**, ID. No. : **2009A3TS005G** in partial fulfillment of the requirements of BITS C421T/422T Thesis embodies the work done by her under my supervision.

Date :

Signature of Supervisor

Dr. Chandra R. Murthy

Assistant Professor, ECE Dept.

Indian Institute of Science

Bangalore - 560 012

ACKNOWLEDGEMENTS

I would like to sincerely thank Dr. Chandra Murthy, Assistant Professor at the ECE Department at the Indian Institute of Science, Bangalore, for giving me the golden opportunity to work on this thesis and for supervising me throughout my project. I am grateful for the continued guidance and support provided by him and for the responsibilities and tasks entrusted to me which served to further my knowledge and technical skills. I would also like to thank my mentor, Mr. Abhay Sharma, for letting me be a part of his work and for all his help, suggestions and ideas in designing the algorithms in this project. I thank all the members of the SPC Lab with all my heart for their valuable inputs and help, for making me feel welcome in their midst and for creating a warm and friendly atmosphere that made my time at IISc so memorable.

Finally, I am extremely grateful to BITS, Pilani University for allowing me to work at IISc for my undergraduate thesis.

Contents

I. Introduction

It was during the second World War that two economists, Robert Dorfman and David Rosenblatt were presented with the problem of testing blood samples of soldiers for syphilis. To perform a blood test on each individual would have been time consuming, and hence, the concept of 'group testing' was introduced that provided an alternative solution.

The basic idea in group testing is to identify a small number of defective items d from a large collection of items N, while minimizing the number of tests required to do so. This is done by creating groups of randomly chosen items from the set and testing a collective property of the group. If the tests are fully reliable then a given test outcome will be negative only if all the items being tested in the corresponding group are non-defective, else it is a positive outcome.

Provided the number of defectives is small, by knowing which groups tested positive for defectives and which ones did not, it is possible to identify all the defective items using a much smaller number of tests in comparison to the number of tests required for testing each item individually. Some applications where group testing is used are : DNA sequencing, testing chemicals, drugs or vaccines for contamination, detection of faulty nodes or links in wireless networks, and many more.

A DNA sequence is stored by cutting it into segments called "clones". The problem is to identify the section of DNA that contains a particular segment called a "probe". The clones containing such probes are the defectives in this case, and are identified using group testing. Two types of tests, "hybridization" or "Polymerace Chain Reaction" can be used on pools of clones. These tests give an observable reaction in case one or more probes are present in the pool, thus telling us that a particular pool tested positive for probes. Using this method, it is possible to identify all clones containing probes, provided they are few in number as compared to the total number of clones being tested.

There are two types of approaches to the group testing problem - 'adaptive' and 'non-adaptive'. In adaptive group testing, the future tests are decided on the basis of outcomes of past tests while in non-adaptive group testing, the pools are decided *a priori* and independently from the outcomes of past tests.

In a compressive sensing problem, we try to recover a sparse signal from a set of incomplete linear measurements. Compressive sensing enables us to sample a signal at sub-Nyquist rates and guarantees its recovery, provided that the signal is sparse (i.e. most of its elements are zero) and that the matrix of linear measurements satisfies certain conditions. Group testing can be interpreted as a compressive sensing problem since the defective items are assumed to be sparse and our goal is to identify them from a large set of items.

The first part of our project involves simulating three optimal non-adaptive group testing algorithms from [1]. Given a certain small probability of error, this paper explicitly calculates upper bounds on the optimal number of tests for each of the three algorithms in both, noisy and noiseless cases. The first algorithm is based on the classical Coupon Collector Problem. The second, called the 'Column Matching Algorithm' is similar to the greedy 'Orthogonal Matching Pursuit' algorithm used in recovery of sparse signals in compressive sensing problems. The third algorithm is based on a convex optimization technique which is novel in its approach since it linearizes the boolean problem of group testing.This paper presents a connection between the fields of compressive sensing and group testing, showing that algorithms designed for the former can also work for the latter. Since a theoretical explanation of the algorithms is presented, it is a worthwhile challenge to practically implement them in MATLAB.

Many times, however, it is more appropriate to identify a subset of "non-defectives" instead of all the defectives. For instance, in a cognitive radio network, a secondary network needs to find a certain amount of free spectrum for setting up secondary operations. In the group testing terminology, the frequency sub-bands used by the primary users are the defective items and the sub-bands not used by the primary users are the non-defective items. Note that the number of free sub-bands (non-defective items) required by the secondary network can be much less than available number of frequency sub-bands. Thus, in this application the main goal is to find an unused chunk of spectrum for the use by the secondary network, i.e., identification of a subset of non-defective items.

The methods used in group testing are an indirect way to find the non-defective items, since they focus on identifying the defective items first, thereby identifying all the non-defective items also.

The next part of our project will be to find methods for identifying a subset of the non-defective items in a non-adaptive group testing setup. Our starting point will be to investigate certain algorithms for non-adaptive group testing for noisy and noiseless cases. For our main problem of directly identifying a subset of non-defectives, we will emphasize on finding methods which are computationally tractable and have provable recovery performance guarantees.

II. Literature Survey

The following papers and articles were referred to while building up the initial understanding of the basic concepts in compressive sensing and group testing :

1) *Richard Baraniuk*, **Compressive sensing**. *(IEEE Signal Processing Magazine, 24(4), pp. 118-121, July 2007)*

   This lecture note describes the inefficiencies of transform coding and the motivation behind the interest in the field of compressive sensing. It discusses the two parts of the problem : design of a stable measurement matrix and the problem of signal recovery. For the first part, coherence and RIP property requirements are briefly enumerated. In the second part, the author considers the l_0, l_1andl_2-norm minimization methods, finally showing that the l_1 norm minimization is the best method since it promotes sparsity. The note ends by citing the example of the single-pixel camera.

2) *Emmanuel Candes and Michael Wakin*, **An Introduction to Compressive Sampling.** *(IEEE Signal Processing Magazine, 25(2), pp. 21 - 30, March 2008)*

   This article describes assertions of Compressive Sensing, the l_1 minimization method for signal recovery (noiseless and noisy) and applications as well as examples.

3) *Richard Baraniuk, Justin Romberg, and Michael Wakin*, **Tutorial on Compressive Sensing** *(2008 Information Theory and Applications Workshop)*

   The tutorial discusses background, basic concepts like RIP, recovery algorithms and tells of applications like single-pixel camera and other imaging applications (Compressive camera array, compressive DNA microarrays)

4) *Mark Davenport, Marco Duarte, Yonina Eldar, and Gitta Kutyniok*, **Introduction to Compressed sensing**, *(Chapter in Compressed Sensing: Theory and Applications, Cambridge University Press, 2012)*

   The document provides a review of vector spaces, and discusses sparse signal models, sensing matrices and the conditions they should satisfy (Null Space Property, Restricted Isometry Property and Coherence) Theorems related to recovery algorithms are also included. This gives an overall picture of the problem and the parts of its solution.

5) *J. A. Tropp and S. J. Wright*, **Computational methods for sparse solution of linear inverse problems** *Proc. IEEE, invited paper, special issue, "Applications of sparse representation and compressive sensing," Vol. 98, num. 5, pp. 948-958, June 2010.*

   This paper discusses various computational methods including the Orthogonal Matching Pursuit (OMP), Compressive Sampling Matching Pursuit(CoSaMP) algorithms and convex optimization algorithms.

6) *Holger Rauhut*, **Compressive Sensing and Structured Random Matrices**, *Radon Series Comp. Appl. Math XX*

   This document contains review of mathematical concepts and tools from probability that will be essential in the understanding of derivations and theorems. It focuses on l_1-minimization method for recovery and structured random matrices.

7) *C. L. Chan, S. Jaggi, V. Saligrama, S. Agnihotri*, **Non-Adaptive Group Testing: Explicit Bounds and Algorithms**, *ISIT 2012*

   This paper draws a parallel between identifying defective items from a set and identifying non-zero elements of a sparse signal. It considers three classes of algorithms for the non-adaptive group testing problem : "Coupon collector Algorithm", "Column Matching Algorithm" and "LP decoding Algorithm" which were inspired by corresponding algorithms in CS literature. The sample-complexity bounds have been derived for all, with constants computed explicitly as functions of desired error probability, noise parameters, number of items and size of defective set (or its upper bound).

III. Compressive Sensing : A Basic Glance

According to the Shannon/Nyquist sampling theorem, a signal must be sampled at atleast twice its bandwidth in order to retain all the information contained in it. However, in numerous applications, the Nyquist rate is too high, thus resulting in a huge amount of captured data, demanding large amounts of memory space for storage. Further, most of the captured data is simply thrown away during the compression stage that follows the acquisition stage. In this way, the traditional method of transform coding (acquire-then-compress) proves inefficient and wasteful.

Compressive sensing is a technique that simultaneously compresses a signal while acquiring it, at a rate well below the Nyquist rate. This is done by taking a small number of linear measurements of the signal, which are used to recover the signal later, by the means of algorithms such as convex optimization, greedy algorithms, etc.

A. Convex Optimization Algorithms

Consider the system of equations

$$\mathbf{y} = \mathbf{A}\mathbf{x} \tag{1}$$

with $\mathbf{x} \in \mathbb{R}^N$, $\mathbf{A} \in \mathbb{R}^{m \times N}$ where $m << N$. This is an underdetermined system, and hence has no solution if the rank of the augmented matrix $[\mathbf{A} \mid b]$ is greater than the rank of the coefficient matrix $\mathbf{A}$. To avoid this case hereafter, we will assume that $\mathbf{A}$ is a full-rank matrix.

To proceed towards solving for $\mathbf{x}$, a natural assumption would be to use the l_2 norm minimization technique. This problem can be represented as follows :

$$(P_2) : \min \|x\|_2^2 \text{ subject to } \mathbf{Ax} = \mathbf{y} \tag{2}$$

This problem can be solved by the **Moore Penrose Pseudoinverse** or the least squares method. But this approach fails to give us a sparse solution as the l_2 norm measures signal energy, and not its sparsity.

The l_0 norm is defined as

$$\|\mathbf{x}\|_0 = \sum |x_i|^0 \tag{3}$$

Hence, this essentially is equal to the number of non-zero elements in $\mathbf{x}$, which in turn is same as the sparsity of $\mathbf{x}$. By minimizing the l_0 norm, we can obtain a suitable sparse recovered vector. The problem of l_0 norm minimization is stated below.

$$(P_0) : \min \|\mathbf{x}\|_0 \text{ subject to } \mathbf{Ax} = \mathbf{y} \tag{4}$$

The above problem is combinatorial and NP-hard, and hence, we do not take this exhaustive approach. Instead, a convex relaxation of (P_0) is the problem of l_1 norm minimization, also known as Basis Pursuit, the solution of which is obtained by well-known algorithms.

$$(P_1) : \min \|\mathbf{x}\|_1 \text{ subject to } \mathbf{Ax} = \mathbf{y} \tag{5}$$

A signal is said to be s-sparse if it has at most s non-zero elements. The above method can exactly recover s-sparse signals and closely approximate compressible signals with high probability using only $m \geq Cs\log(N/s)$ measurements (refer [2], [3]) where N is the length of the signal.

This is known as a convex optimization problem. The complexity of the Basis Pursuit algorithm is $O(N^3)$.

B. *Greedy Algorithms*

Greedy algorithms iteratively improve a sparse solution by successively identifying one or more coefficients that yield the greatest refinement in the estimation of the desired signal $\mathbf{x}$. Examples of greedy algorithms are the Orthogonal Matching Pursuit(OMP)and its variants like the Compressive Sampling Matching Pursuit (CoSaMP). One of the relevant documents this subject is [8]. The CoSaMP algorithm is a greedy method that incorporates ideas from the combinatorial algorithms to guarantee speed and to provide rigorous error bounds. Greedy pursuits are intermediate in their running time and sampling efficiency as compared to convex or combinatorial methods. An excellent reference for the CoSaMP is [11].

IV. Simulations using l_1 Magic

l_1-MAGIC is a collection of MATLAB routines for solving the convex optimization programs central to compressive sampling. The algorithms are based on standard interior-point methods and were implemented by Emmanuel Candes and Justin Romberg at Caltech.

Some simulations were carried out using these codes during the literature survey phase of this project, for an understanding of the optimization techniques involved. The results obtained were as follows.

A. Min. l_1 with Equality Constraints

This program finds the vector with minimum l_1 norm for the problem:

$$P_1 : \min \|\mathbf{x}\|_1 \text{ subject to } \mathbf{Ax} = \mathbf{y}$$

The code implements the primal-dual interior point algorithm. It creates a sparse signal, takes a limited number of measurements of that signal, and recovers the signal exactly by solving (P_1). The measurements y are taken, and the “minimum energy” solution is calculated (shown in Figure 2) . This is the vector in $\mathbf{x}$:$\mathbf{Ax}$ = $\mathbf{y}$ that is closest to the origin.

The inputs to the algorithm are :

1) Initial guess for the solution (x0)
2) Measurement matrix ($\mathbf{A}$)
3) The measurements ($\mathbf{y}$)
4) Precision to which we want the problem solved (1e-3)

B. Min. l_1 with Quadratic Constraints

This program finds the vector with minimum l_1 norm for the problem:

$$P_2 : \min \|\mathbf{x}\|_1 \text{ subject to } \|\mathbf{Ax} - \mathbf{y}\|_2 \leq \epsilon$$

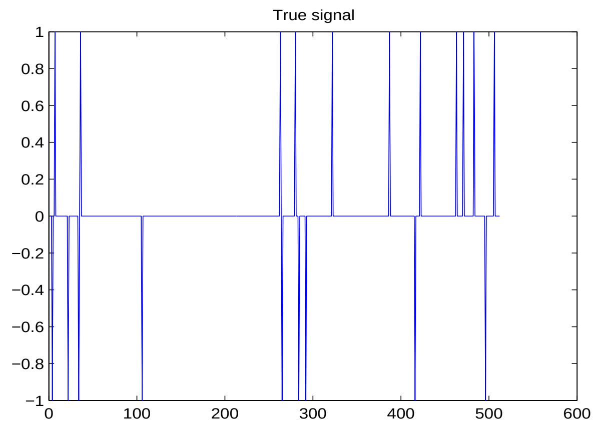

Fig. 1. Min. l_1 with Equality Constraints : True sparse signal $\mathbf{x}$ consisting of 20 spikes

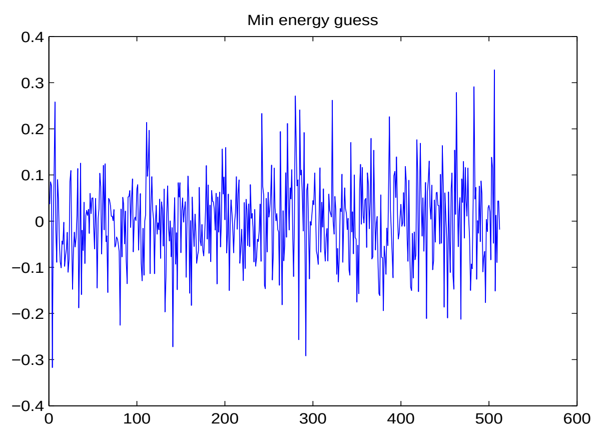

Fig. 2. Min. l_1 with Equality Constraints : Initial guess (minimum energy) reconstruction

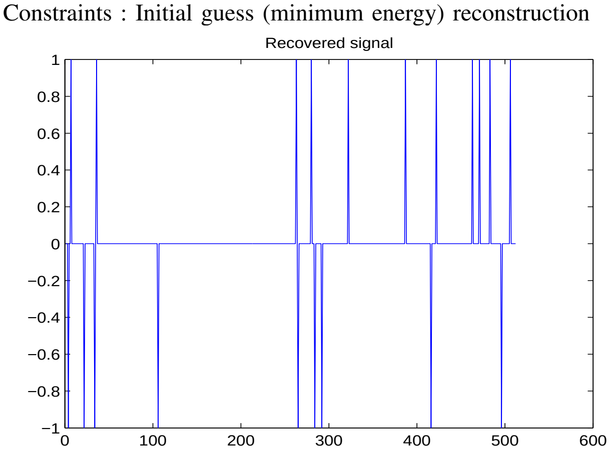

Fig. 3. Min. l_1 with Equality Constraints : Recovered Signal

Here ϵ is a user-specified parameter. For some small error term, $\|e\|_2 \leq \epsilon$.

The log barrier algorithm is used in this case. The procedure is similar to the above approach, and an initial guess solution is calculated. The inputs to the algorithm are :

1) Initial guess for the solution (x0)
2) Measurement matrix ($\mathbf{A}$)
3) The measurements ($\mathbf{y}$)
4) Precision to which we want the problem solved (1e-3)

These proof-of-concept codes show that the recovery procedures are computationally tractable, even for large scale problems. This simulation exercise provided an understanding of convex optimization methods used in compressive sensing.

V. Non-Adaptive Group Testing : Algorithms

A. *Coupon Collector Algorithm*

The Coupon Collector's algorithm (**CoCo**) is based on the classical Coupon Collector problem. This problem involves sampling with replacement from a set of N types of coupons, requiring us to find a bound on the minimum number of draws in which all N types of coupons will be collected. In the context of group testing, the goal of **CoCo** is the collection of all the non-defectives.

This is a *row-wise* algorithm. Figure 7 is an example demonstrating the working of this method.

The group-testing matrix $\mathbf{A}$ is a $T \times N$ matrix, where T is the number of tests and N is the total number of items. The entries of $\mathbf{A}$ are decided in the following manner : for the i^{th} row entries, we sample with replacement from the set $[N]$ exactly g number of times (g is known as the *group sampling parameter.*) We set $\mathbf{A}(i, j)$ equal to one if j is included in the samples at least once. Else, $\mathbf{A}(i, j)$ is set to zero.

In order to recover the vector $\mathbf{x}$ i.e. the information vector which identifies the defectives, we use the tests which have a negative outcome. All the items included in tests having a negative outcome are declared non-defective. The remaining items are classified as the defectives.

Provided that there are enough number of tests and the number of defectives is much smaller than the total number of items, [1] claims that this algorithm is effective. However, two scenarios in which errors may occur are : (1) at least one non-defective item is not tested at all or (2) at least one non-defective item appears only in tests with positive outcomes.

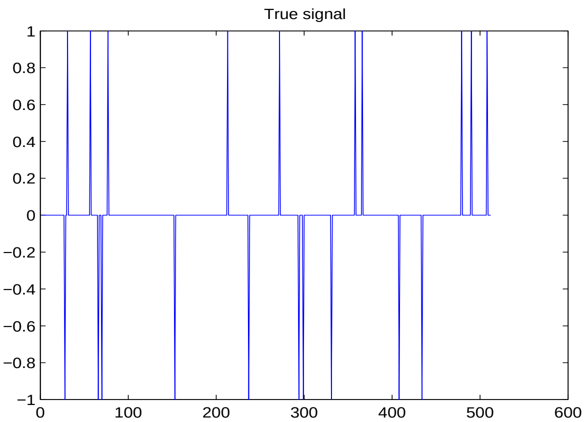

Fig. 4. Min. l_1 with Quadratic Constraints : True sparse signal $\mathbf{x}$ consisting of 20 spikes

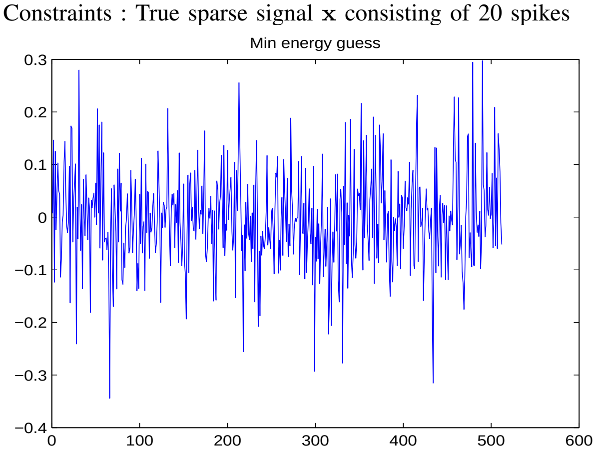

Fig. 5. Min. l_1 with Quadratic Constraints : Initial guess (minimum energy) reconstruction

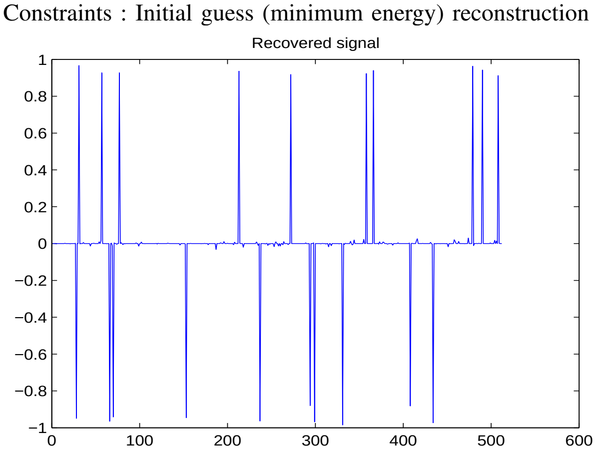

Fig. 6. Min. l_1 with Quadratic Constraints : Recovered Signal

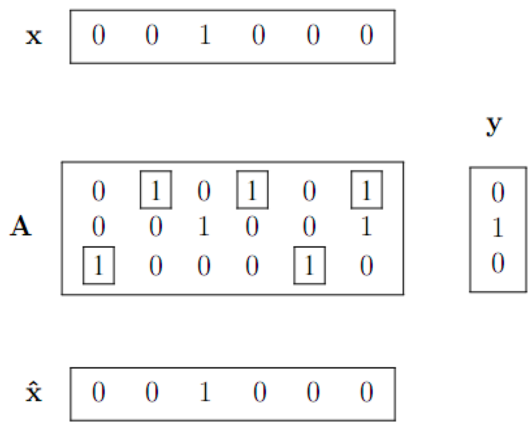

Fig. 7. The set up of the **CoCo** algorithm. $\mathbf{x}$ is the boolean "information vector" (unknown to us) : if x_j = 1, the j^{th} item is defective, else it is non-defective. $\mathbf{A}$ is the group-testing matrix and $\mathbf{y}$ is the matrix of test outcomes. $\hat{\mathbf{x}}$ is the recovered vector.

B. Column Matching Algorithm

The Column Matching algorithm has two cases - noiseless (**CoMa**) and noisy (**NoCoMa**). In the latter case, a noise vector gets added to the vector of test outcomes. This algorithm is a *column-wise algorithm* since it correlates the columns of matrix $\mathbf{A}$ with $\mathbf{y}$.

The elements of the $T \times N$ matrix $\mathbf{A}$ are drawn i.i.d. and $\mathbf{A}(i, j)$ is one with probability p. The algorithm attempts to find the columns of $\mathbf{A}$ which are "contained" in $\mathbf{y}$. The elements corresponding to such columns are clearly the defectives.

Note that the noiseless column matching algorithm (Figure 8) can result in classifying some items as false defectives, but will never result in classifying defectives as non-defectives. The false defective classification may occur if a non-defective item is hidden by defectives occurring in the same tests.

For the noisy case (Figure 9), we allow for certain number of "mismatches" in the locations of ones. The number of mismatches depends on the noise parameter q (the probability by which the test outcome differs from the true outcome) as well as the number of ones in a column. Let T_j be the indicator set i.e set of indices i such that $\mathbf{A}(i, j)$ = 1. Let S_j be the "matching set", or the set of indices j such that $\hat{y}_i$ = 1 and $\mathbf{A}(i, j)$ = 1. The thresholding rule used in this paper is such that if $|S_j| \geq |T_j| \left(1 - q(1 + \tau)\right)$ then the j^{th} item is classified as defective. Note that τ is a design parameter.

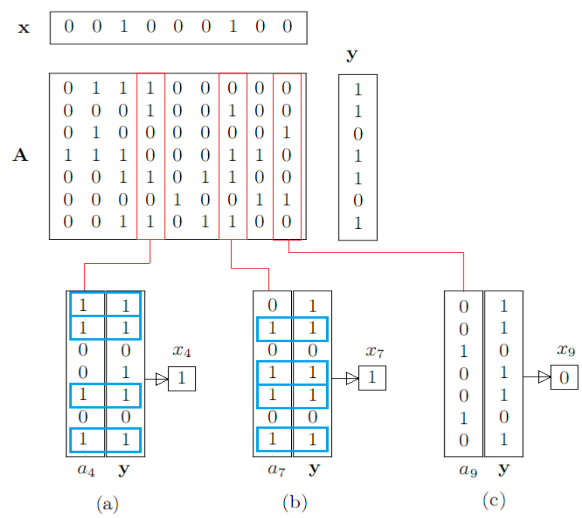

Fig. 8. The set up of the **CoMa** algorithm. $\mathbf{x}$ is the boolean "information vector" (unknown to us). $\mathbf{A}$ is the group-testing matrix and $\mathbf{y}$ is the matrix of test outcomes. $\hat{\mathbf{x}}$ is the recovered vector. The algorithm matches columns of $\mathbf{A}$ to $\mathbf{y}$. For (b), the 7^{th} column is contained in $\mathbf{y}$, hence $x_7 = 1$. In (c), the item is seen to be non-defective, hence $x_9 = 0$. There is an error in (a) as the non-defective item is hidden by columns corresponding to other defectives.

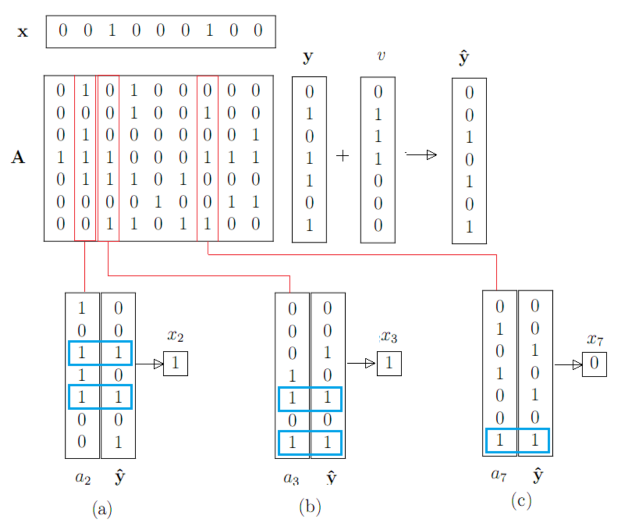

Fig. 9. The set up of the **NoCoMa** algorithm. The algorithm matches columns of $\mathbf{A}$ to $\mathbf{y}$ while allowing a certain number of mismatches, set by a threshold. Here, we set the threshold such that number of matches is more than number of mismatches. For (b), the item is correctly identified as defective. In (c) however, we have an error due to noise, generating a false non-defective. There are measurement errors in (a) as well, leading to a false defective classification.

C. LP-decoding algorithms

The linear programming decoding algorithms in [1] are described for both noisy and noiseless scenarios, and for the following two cases as well :

1) The number of defective items d is exactly known.
2) Only the upper bound on the number of defectives D is known.

The $T \times N$ group testing matrix $\mathbf{A}$ is drawn i.i.d. by setting each entry to one with a probability $p = 1/D$ or $p = 1/d$, whichever is the appropriate setting. Although the group testing problem is boolean, this algorithm relaxes it into a linear problem by not restricting x_j to take on the values 0 or 1 alone. Instead, $x_j \in$[0, 1], otherwise the problem becomes NP-hard. The noiseless linear programming problem is stated below.

$$\hat{\mathbf{x}} = \text{feasible point in} \tag{6}$$

$$\sum_{j:\mathbf{A}(i,j)=1} x_j = 0 \text{ if } y_i = 0, \tag{7}$$

$$\sum_{j:\mathbf{A}(i,j)=1} x_j \geq 1 \text{ if } y_i = 1, \tag{8}$$

$$\sum_{\forall j} x_j = d \tag{9}$$

$$0 \leq x_j \leq 1 \tag{10}$$

This method, called **LiPo**, outputs the feasible solution for a given value of d. In the presence of noise, the algorithm is modified as follows.

$$(\hat{x}, \hat{\eta}) = \min \sum_{y_i=1} \eta_i + \frac{1}{e} \sum_{y_i=0} \eta_i \tag{11}$$

such that

$$-\eta_i + \sum_{j:m_{i,j}=1} x_j = 0 \text{ if } \hat{y}_i = 0 \tag{12}$$

$$\eta_i + \sum_{j:m_{i,j}=1} x_j \geq 1 \text{ if } \hat{y}_i = 1 \tag{13}$$

$$\sum_{\forall j} x_j = d \tag{14}$$

$$0 \leq x_j \leq 1 \tag{15}$$

$$0 \leq \eta_i \leq d \text{ if } \hat{y}_i = 0 \tag{16}$$

$$0 \leq \eta_i \leq 1 \text{ if } \hat{y}_i = 1 \tag{17}$$

The slack variable η accounts for errors in the test outcomes. When a test is truly negative, all the terms in equation (12) are zero. If it is a false negative test however, the slack variable is set to be equal to the number of defective items in that test. Similarly, if a test is a false positive, the slack variable is set to 1. This is an asymmetry that results from the fact that multiple defective items in a test give the same result as a single defective item included in a test. Here, we implement a "minimum distance decoding" criteria. We decode the pair $(\mathbf{x}, \eta)$ such that the error-vector is as small as possible.

Now, we account for the case in which only the upper bound on the number of defectives is known. This variation of the above algorithm is called the Noisy Universal Linear Program (NoUnLiPo) and the optimization problem is defined as follows.

$$\forall \bar{d} \in \{0, \cdots D\}, (\hat{x}(\bar{d}), \hat{\eta}(\bar{d})) = \arg\min_{x,\eta} \sum_i \eta_i \tag{18}$$

such that

$$-\eta_i + \sum_{j:m_{i,j}=1} x_j = 0 \text{ if } \hat{y}_i = 0 \tag{19}$$

$$\eta_i + \sum_{j:m_{i,j}=1} x_j \geq 1 \text{ if } \hat{y}_i = 1 \tag{20}$$

$$\sum_{\forall j} x_j = \bar{d} \leq D \tag{21}$$

$$0 \leq x_j \leq 1 \tag{22}$$

$$0 \leq \eta_i \leq \bar{d} \text{ if } \hat{y}_i = 0 \tag{23}$$

$$0 \leq \eta_i \leq 1 \text{ if } \hat{y}_i = 1 \tag{24}$$

This method outputs the first feasible solution obtained by running the LP, starting sequentially from $\bar{d} = 0$. The decoder increments $\bar{d}$ by 1 till it finds a valid solution. If $\bar{d} > D$ and no solution is found, the decoder declares that an error has occurred.

VI. Upper Bounds on number of tests

[1] contains explicitly computed results on the number of tests required for a desired error probability via computationally efficient algorithms. These results are presented in the form of theorems, and the understanding of their proofs has been one of the tasks in this project. The proofs have been presented in the appendix. For now, the theorems stated in the paper have been restated here, and are as follows.

Theorem 1: **CoCo** with error probability at most $n^{-\delta}$ requires no more than $2(1+\delta)eD\ln N$ tests.

Theorem 2: **CoMa** with error probability at most $n^{-\delta}$ requires no more than $eD(1+\delta)\ln(N)$ tests.

Theorem 3: **NoCoMa** with error probability at most $n^{-\delta}$ requires no more than $\dfrac{16(1+\sqrt{\gamma})^2(1+\delta)ln\,2}{(1-e^{-2})(1-2q)^2}D$ log(N) tests.

Note that the thresholding performed in the case of **NoCoMa** leads to the above result. The constants are defined as given below :

- $\gamma = \frac{\Gamma+\delta}{1+\delta}$. Hence $\gamma \in (\delta/(\delta+1), 1]$.
- $\Gamma = \frac{\ln(D)}{\ln(N)}$, $\Gamma \in [0,1)$
- $\tau = \frac{1-2q}{q(1+\gamma^{-1/2})}$
- $p = \frac{1}{D}$

Theorem 4: **LiPo** with error probability at most $n^{-\delta}$ requires no more than $\beta_{LP}D\ln(N)$ tests.

Here β_{LP} is $\max\{\frac{4e(\delta+1+\Gamma)}{(1-2q)^2}, 8e(\delta+1+\Gamma), \frac{4e(1-q+2qe)(\delta+1+\Gamma)}{(1-q)^2}, \frac{8e(\delta+1+\Gamma)}{(1-q+2qe)}, \frac{(1-q+qe)(\delta+\Gamma)(1+e)^2}{e(1-2q)^2}, \frac{8e(\delta+\Gamma)}{(1-q+qe)}\}$

VII. Simulations

A. Coupon Collector Algorithm

For the implementation of **CoCo**, we considered three cases:

1) N = 256, d = 8
2) N = 512, d = 16
3) N = 1024, d = 32

We generate the information vector **x** randomly such that it contains the required number of defectives. To generate the i^{th} row of $\mathbf{A}$, we sample randomly g times from the set $[N]$, with replacement. If the index j appears atleast once in the g samples, the element $\mathbf{A}(i, j)$ is set to one, else it is zero.

Next, we run the algorithm a fixed number of times for T number of tests, where T varies linearly. The items appearing in negative tests are declared non-defective and those not appearing in any negative tests are classified as defectives. The probability of error is calculated, and hence, the probability of recovery, which is one minus the former. This is plotted versus the number of tests divided by a factor of $d\log(N)$. The plot is as shown in Figure 10.

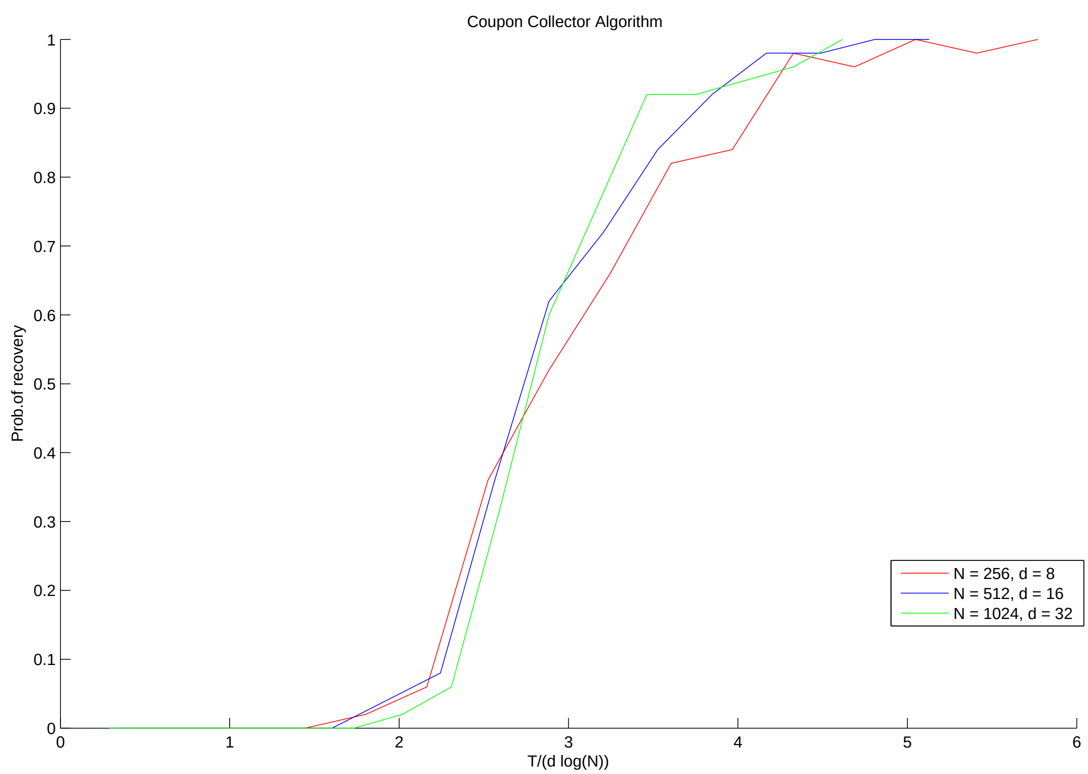

Fig. 10. Coupon Collector Algorithm : Probability of Recovery

By substituting the value of probability of recovery as 0.8, and hence probability of error as 0.2, we get the estimates for upper bounds from [1].

Our goal is to show that the upper bound on the number of tests scales as $d\log(N)$. We therefore plot probability of recovery with number of tests required, as seen in Fig. 12. Here, for probability of recovery equal to 0.8, we note the values of tests required for each pair (N, d) (Table 13) We would

N	d	Theoretical β	Theoretical Upper Bound	Actual β (graph)
256	8	48.84	2167	3.562
512	16	49.26	4916	3.422
1024	32	49.56	10993	3.252

Fig. 11. Comparison between simulated and theoretical results. According to Theorem 1, the upper bound for a desired error probability of $N^{-\delta}$ is $\beta d\log(N)$, where β is as given in the theorem. Here we find that the actual values of β are less than the theoretical ones.

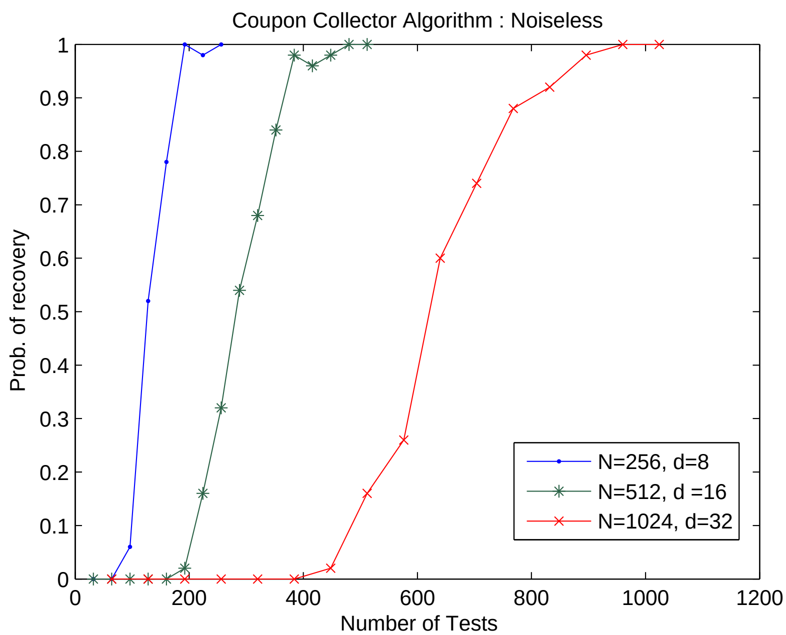

Fig. 12. Prob. of recovery vs. Number of tests required for **CoCo**

N	d	Number of tests
256	8	163
512	16	344
1024	32	731

Fig. 13. For probability of recovery = 0.8, or probability of error $N^{-\delta}$ = 0.2, the number of tests required for each pair (N, d) for **CoCo**.

like the number of tests to scale as $d\log(N)$ i.e. the plot of the number of tests vs. $d\log(N)$ must be a straight line, which we see in Fig. 14, it is.

B. Column Matching Algorithm

1) Noiseless case - CoMa: To correlate the columns of the measurement matrix with $\mathbf{y}$, for each column j, we check whether the dot product of column a_j and $\mathbf{y}$ equals the number of ones in a_j. If this matches, we classify the j^{th} item as defective.

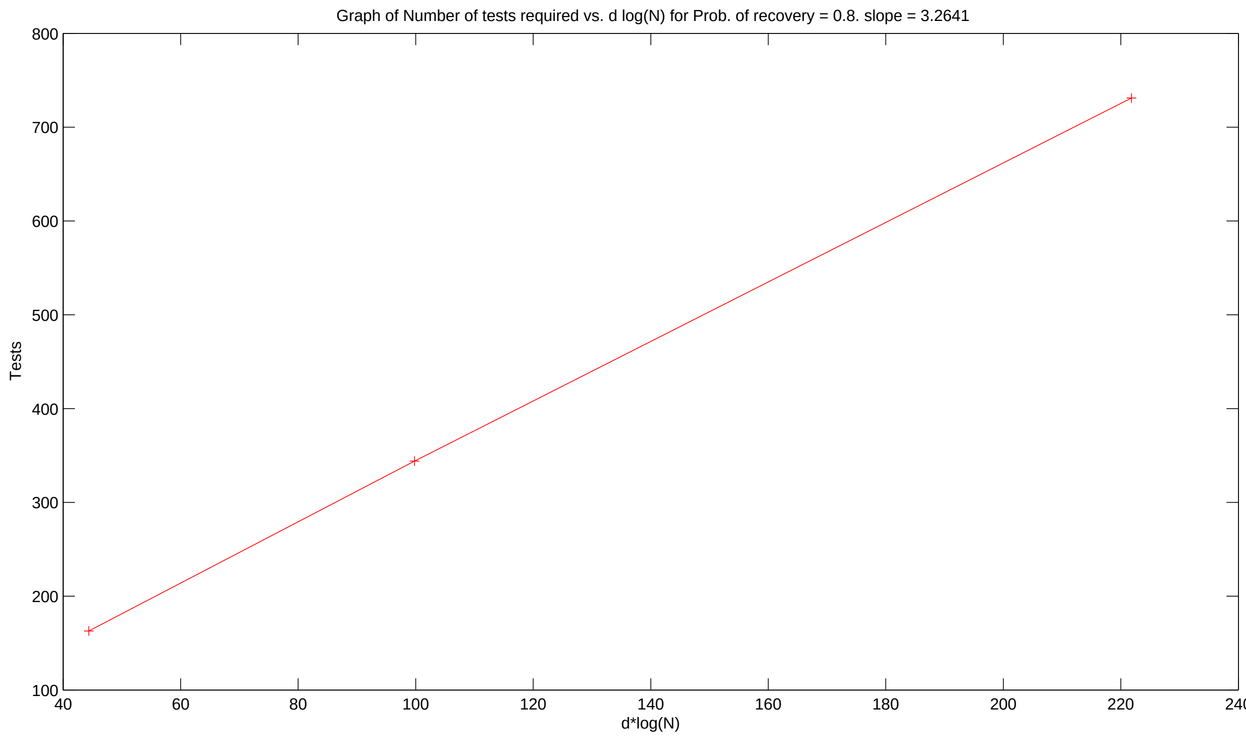

Fig. 14. This plot shows that the upper bound for the Coupon Collector Algorithm does scale as $d\log(N)$. The slope of the line is 3.2641

In Fig. 15, we plot probability of recovery on the Y-axis and $T/(d\log(N))$ on the X-axis. It is seen that the curves for different values of N and d lie on top of one another, implying that the upper bound on the number of tests for a given probability of recovery in fact, does scale as $d\log N$.

Also, by subsitution of probability of recovery equal to 0.8, we see that the obtained bounds are comparable to the theoretical values. (Table 16).

Analyzing this data as was done for **CoCo**, for probability of recovery equal to 0.8, we note the values of tests required for each pair (N, d) (Table 17) Again, we see that the plot of the number of tests vs. $d\log(N)$ is nearly a straight line, in Fig. 18.

2) Noisy case - NoCoMa: Let us consider the noisy case, where a noise vector is logically OR'ed with the vector of test outcomes $\mathbf{y}$. Note that q_0 is the probability that a test result differs from the true result.

Here, instead of *exactly* matching the dot product of column a_j and $\mathbf{y}$ with the number of ones in a_j, we set a certain threshold and allow for upto two mismatches. For q_0 = 0.2, and probability of recovery equal to 0.7, we get Fig. 19, which shows that the number of tests scale linearly with d log N, as

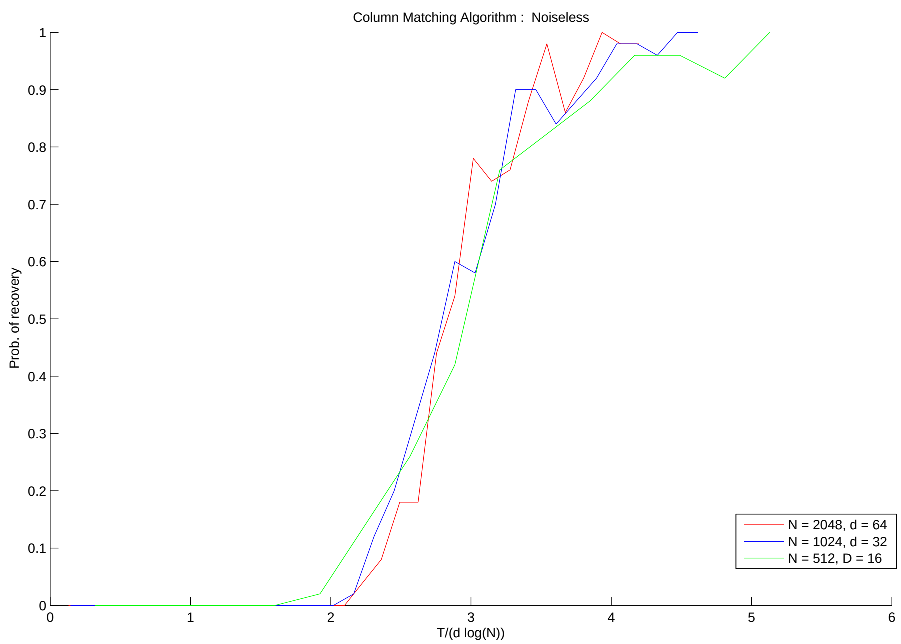

Fig. 15. Noiseless Column Matching Algorithm

N	d	Theoretical β	Theoretical Upper Bound	Actual β (graph)
512	16	3.4192	341	3.42
1024	32	3.3491	742	3.246
2048	64	3.2917	1606	3.323

Fig. 16. Comparison between the theoretical and simulated results for the **CoMa** algorithm shows that the actual values of number of tests required lie within the proposed bounds.

N	d	Number of tests
512	16	352
1024	32	772
2048	64	1590

Fig. 17. For probability of recovery = 0.8, or probability of error $N^{-\delta}$ = 0.2, the number of tests required for each pair (N, d) for **CoMa**.

claimed.

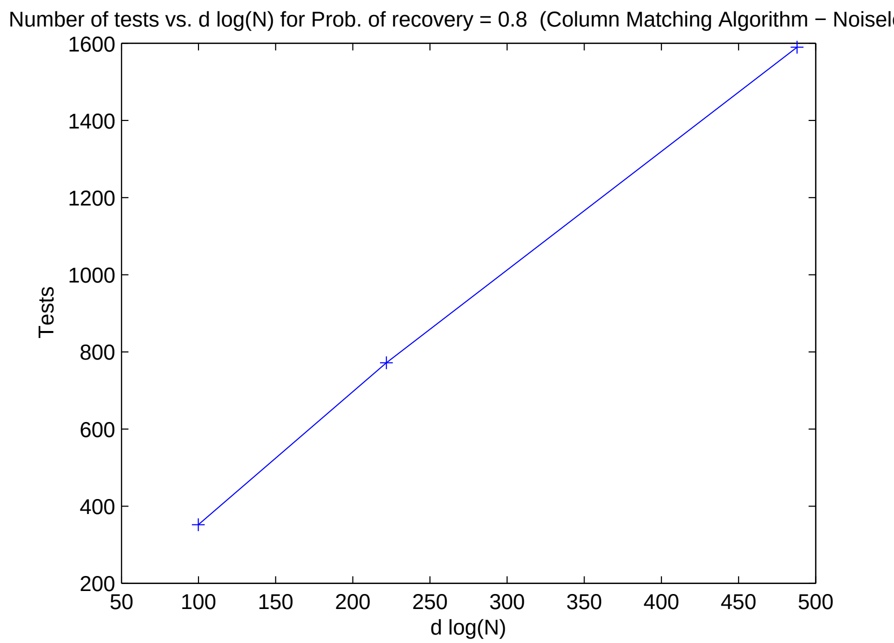

Fig. 18. This plot shows that the upper bound for the **CoMa** Algorithm does scale as $d\log(N)$. The slope of the line is 3.4428, which is also near to the theoretical values of β found previously.

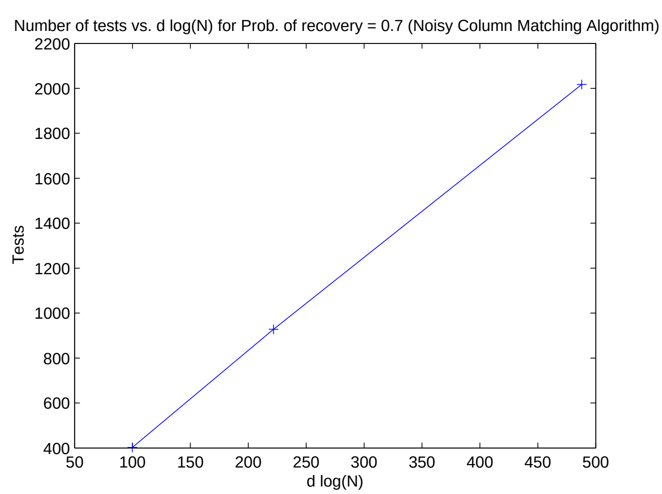

Fig. 19. The upper bound scales linearly with $d \log N$ for the Noisy Column Matching Algorithm as well.

C. LP Decoding Algorithm

Using the linear program given in [1], the MATLAB simulation yields the graph in Fig. 20 for Probability of recovery vs. number of tests.

Note that the values of x_i obtained through the linear program lie in [0,1], as we have relaxed the constraint. Hence, once the solution to the LP is available, we pick the topmost d entries and classify them as defectives. This can lead to errors, but as the number of tests increases, the probability of errors reduces, as seen from the graphs.

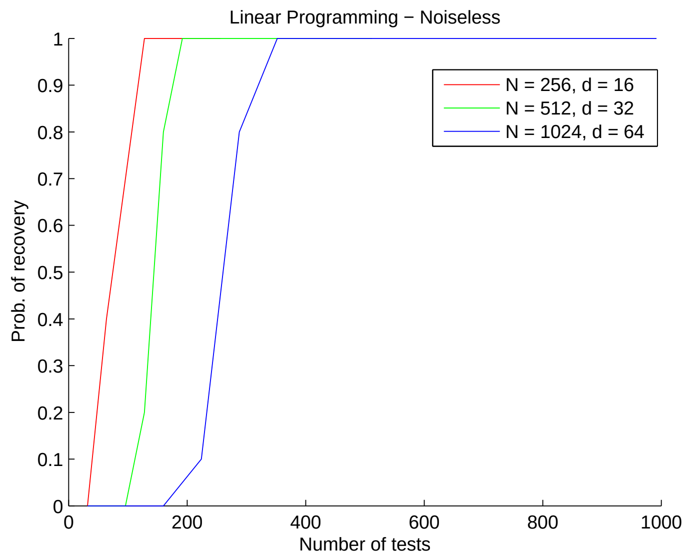

Fig. 20. Prob. of recovery vs. T, plotted for different values of defectives d and total items N

For probability of recovery equal to 0.9, we plot the number of tests on the Y-axis and d log$N$ on the X-axis, in the hope of verifying the bound as given in the paper. Here too, we see that the plot is a straight line (Fig. 21) This verifies the claim in the paper.

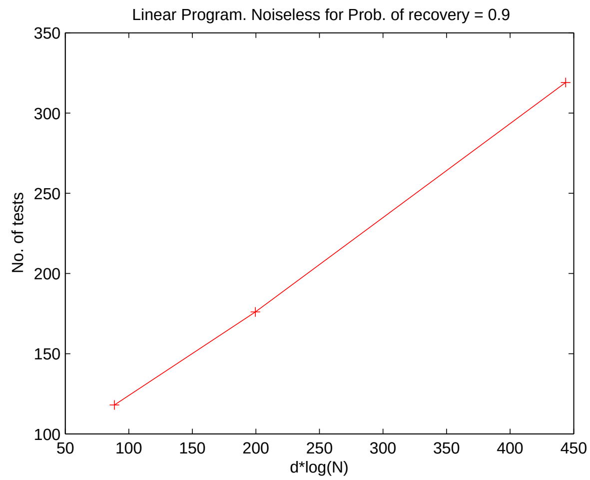

Fig. 21. For the Noiseless LP Decoding algorithm, it is seen that the bound scales linearly with $d\log(N)$

Now, for the noisy case, we simulate additive noise with probability 0.05. The probability of recovery is plotted as shown in Fig. 22. For probability of recovery equal to 0.9, we see that the graph of Number of tests vs. d log$N$ is almost a straight line in Fig. 23. Also note that the number of tests for the noisy case is much more than that required for the noiseless case.

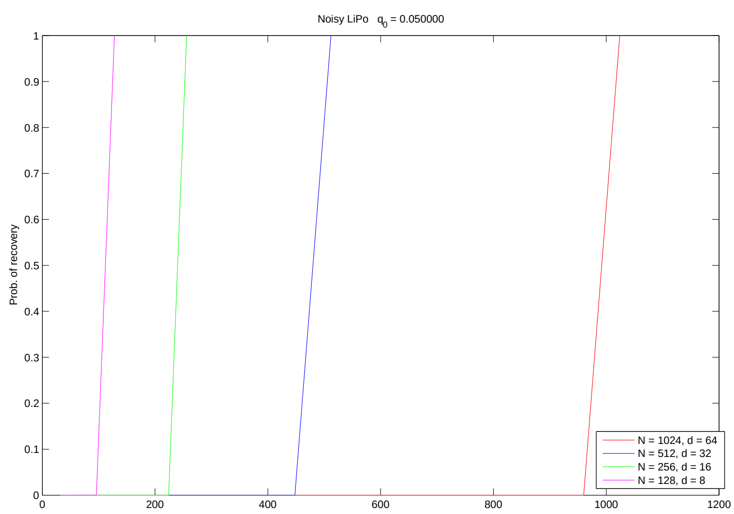

Fig. 22. Prob. of recovery vs. T, plotted for different values of defectives d and total items N for Noisy LP Decoding Algorithm

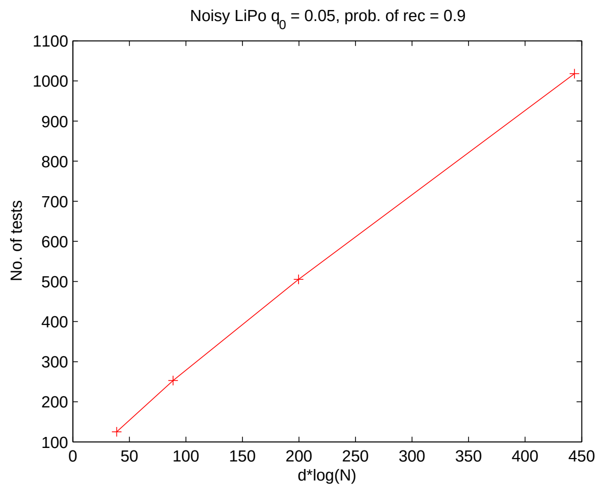

Fig. 23. For the Noisy LP Decoding algorithm, it is seen that the bound scales linearly with $d\log(N)$

VIII. The Finding Zeros Problem

Now we arrive at the main focus of the project : the finding zeros problem. Since “zeros” are essentially the non-defectives in a given population, we will use these terms alternatively. Sometimes, it is more important to find a small subset of the non-defective items or “zeros” rather than locating all the defectives. An example is the cognitive radio network, where it is more important to find a free chunk of spectrum rather than to locate all the primary stations. The methods used in group testing are an indirect way to find the non-defective items, as they identify all defective items first, thereby identifying all the non-defective items also.

This takes a large number of tests, however, and in this work we will show that our algorithms can identify a subset of zeros using a much smaller number of tests. These algorithms are modifications of the three group testing algorithms discussed previously. Let us consider them one by one.

A. Coupon Collector - Based Approach

For the problem of identifying a subset of zeros, the modification in the Coupon Collector Algorithm is as follows:

- Count the number of repetitions of each item j over all the tests that gave a negative outcome.
- If an item never appears in any negative tests, we cannot conclude anything about it : it could be a defective item, or a non-defective item which did not appear in any negative tests.
- Hence, we count the items which appear at least once for a particular number of tests, and atleast twice in case of additive noise, for a fixed error probability.

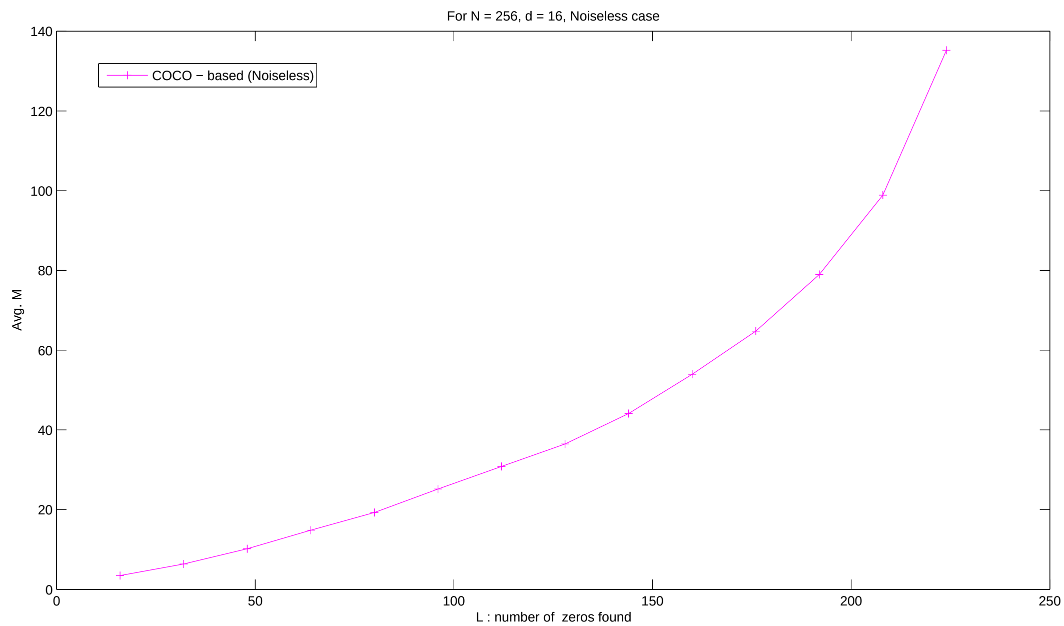

Fig. 24. For the Coupon Collector - based approach, average number of measurements required to find L non-defectives is obtained from these results.

Using this algorithm, we plot the average number of tests required to find a subset of zeros of size L, on the Y-axis, with L on the X-axis.

Fig. 24 was for the noiseless case. As we can see, the number of measurements is much lower than what we would require if we were using one of the group testing algorithms or performing one-by-one testing.

Consider the two kinds of noise that may enter our setup : additive noise and dilution noise.

- Additive model: False alarms could arise from errors in some of the tests. This happens when some tests are erroneously positive.
- Dilution model: Even though a positive item is contained in a given pool, the test's outcome could be negative if the defective item gets diluted for that specific test. For example, in blood testing the positive sample might get diluted in one or more tests leading to potential misses of infected blood samples.

We assume that the probability that an entry gets flipped is q. In this modified algorithm, consider the case of additive noise. For the noisy case, the items which are repeated more than r number of times are classified as defectives since the probability that a positive test outcome gets flipped to negative due to noise is small. Hence, the probability that defective items are classified as non-defectives over and

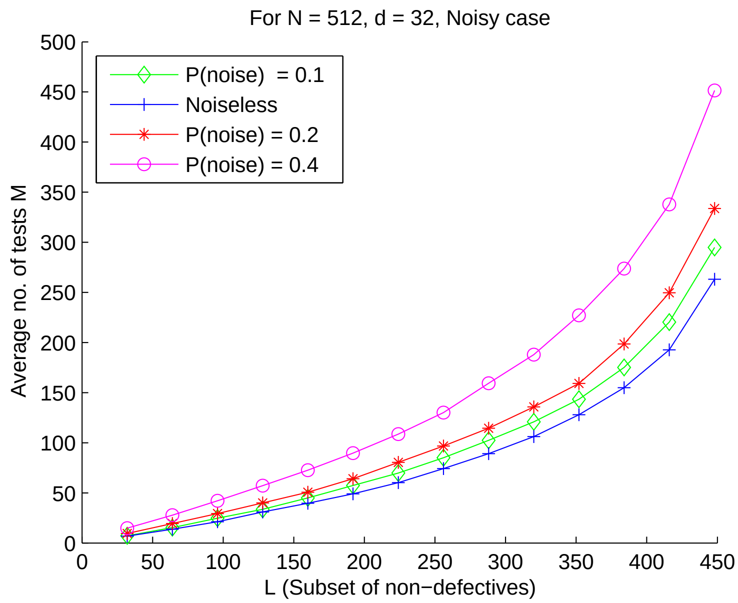

Fig. 25. For the Coupon Collector - based approach, average number of measurements required to find L non-defectives : comparison for different noise probabilities

over again is very small. It is found that the probability of error in the algorithm is very low or zero for $r = 2$. Hence, this value of r is sufficient, and the graphs of average number of measurements required vs. subset of zeros found are as shown in Fig. 25, for different probabilities of additive noise. We see that as noise increases, we require more tests to find the subset of zeros correctly, as is expected.

B. Column Matching - Based Approach

This approach is similar to the concept of the Orthogonal Matching Pursuit. In OMP, we choose columns with highest inner product with the result vector y. Here, to obtain a set of L zeros, the L columns of A which give the least inner product with y are chosen. Hence, the algorithm computes the inner product of each column with y and then arranges these inner products in ascending order. The top L entries are picked and classified as zeros.

The graph of average number of measurements vs. subset of zeros is as seen in Fig. 26. Again, we see that this is much better than the group testing algorithm if we require a subset of zeros to be found instead of the defective items, or ‘ones’.

C. LP - Based Approach

The linear program from the group testing setup is modified to suit our needs of finding a subset of zeros. Here, we denote $\mathbf{z} = 1 - \mathbf{x}$, where $\mathbf{x}$ was the information vector we were trying to find. In $\mathbf{z}$, the

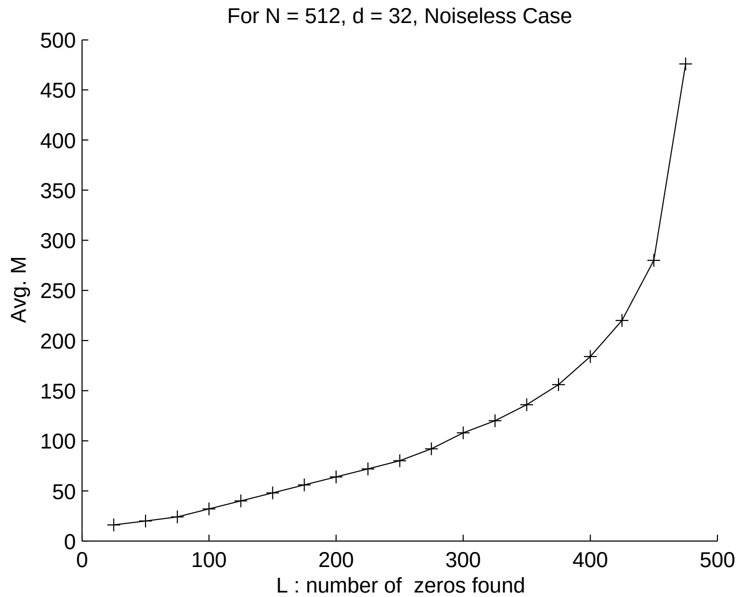

Fig. 26. For the noiseless Column Based Approach to finding zeros, average number of measurements required to find L non-defectives.

entries which are one denote the non-defective items. The modified linear program is as follows :

$$\min \sum_{y_i=1} \eta_i + \frac{1}{e} \sum_{y_i=0} \eta_i \tag{25}$$

such that

$$-\eta_i + \sum_{j:a_{i,j}=1} (1 - z_j) = 0kj \text{ if } \hat{y}_i = 0 \tag{26}$$

$$\eta_i + \sum_{j:a_{i,j}=1} (1 - z_j) \geq 1 \text{ if } \hat{y}_i = 1 \tag{27}$$

$$\sum_{\forall j} z_j \geq L \tag{28}$$

$$0 \leq z_j \leq 1 \tag{29}$$

$$0 \leq \eta_i \leq d \text{ if } \hat{y}_i = 0 \tag{30}$$

$$0 \leq \eta_i \leq 1 \text{ if } \hat{y}_i = 1 \tag{31}$$

The condition $\sum_{\forall j} z_j \geq L$ allows us to obtain the subset of zeros of size close to L. Running this algorithm for different values of L, the Y-axis in Fig. 27 represents the number of tests that found L zeros without errors. This is the noiseless case. The noisy cases remain to be investigated.

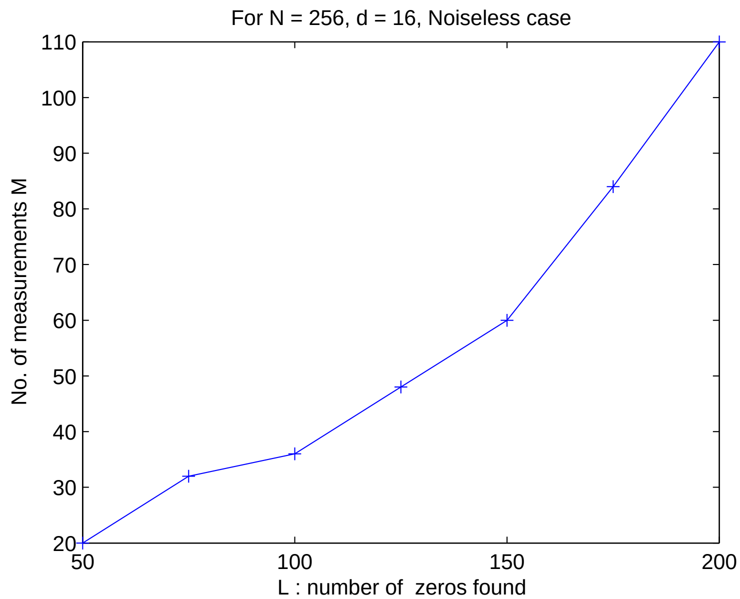

Fig. 27. For the noiseless LP Approach to finding zeros, this graph depicts number of measurements required to find L non-defectives. It was obtained by running the LP for all values of measurements M and a particular value of L, and picking the M which gave less than 1% probability of error.

D. Comparison of Performances

Comparing the graphs for all the three methods, in Fig. 28, we see that the Coupon-Collector based method performs best in the noiseless case. The Column Matching Based method is the next best, followed by the LP - Based method. Fora each of the methods, we generate an array of number of zeros found (L) for a particular number of measurements (M). For a point L on the X-axis, the corresponding point on the Y-axis is the value of M that yields L or more zeros with a fixed high probability. In case of the linear program, we run the program for a given value of L and various values of M and pick the M that gives less than 1% probability of error.

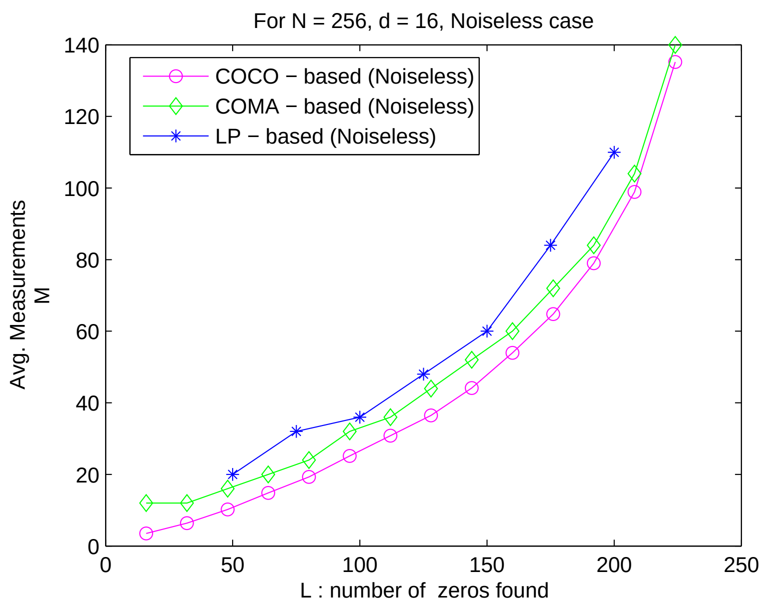

Fig. 28. Comparison of the number of tests required for each of the proposed methods for the problem of finding L zeros in a given population with a fixed, known number of defectives (ones).

IX. Conclusion

In this thesis, a study of the basics of compressive sensing was done, along with a literature survey. The algorithms in [1] were implemented and analyzed for the noiseless as well as the noisy cases. The upper bounds on the number of tests required for each of these group testing algorithms were verified through MATLAB simulations. Importantly, the problem of finding a subset of non-defectives was considered. The three algorithms mentioned in [1] were modified to solve this new problem. The number of tests required to find a subset of non-defectives using the modified versions of the Coupon Collector Algorithm, the Column Matching Algorithm and the LP Decoding Algorithm were found to be much smaller than the number of tests required by the original algorithms (which identify the defectives first and enable us to pick the non-defectives from the remaining items). The noiseless case for all three modified methods was compared, and it was found that the Coupon Collector - based approach performed best, followed by the Column Matching - based approach and the LP - based approach respectively. The noisy cases remain to be investigated, but will be analyzed as a part of future work on this problem. We will also come up with provable guarantees for the same, and compare the results to find the most optimal algorithm in both noiseless and noisy scenarios.

APPENDIX

A. Proofs of performance

Let us now consider the proofs of the performance of the algorithms. Consider the **CoCo** first.

1) Coupon Collector Algorithm: The classical Coupon Collector's problem has an expected stopping time of $N\ln(N) + \Theta(N)$, according to [10]. It is also known that the probability that the stopping time is more than $\chi N\ln(N)$ is at most $N^{-\chi+1}$.

Note that the probability that a particular item appears in the g-length sampled vector in any location is uniform and independent, even across tests. Hence, selecting items by performing k tests and sampling g times in each test is the same as selecting a chain of gk coupons. Since the aim of this algorithm is to collect all the non-defectives, we can write

$$Tg\left(\frac{N-d}{N}\right)^{g} \geq (N-d)\ln(N-d) \tag{32}$$

Here, the left hand side is the expected number of items in negative tests, including possibly repeated. This is derived from the fact that there are T tests, each containing g items, some of which may be repeated. Also, the probability that a test is negative is $\left(\frac{N-d}{N}\right)^{g}$. The right hand side of (32) is the expected stopping time of the classical Coupon Collector algorithm.

On optimizing (32) with respect to g, we get

$$\begin{aligned} g &= -\frac{1}{\ln(N-d/N)} \\ &= \frac{1}{\ln(N/N-d)} \end{aligned}$$

Since the exact value of d may not be known, but the upper bound on it D is known, we set

$$g = \frac{1}{\ln(N/N-D)} \tag{33}$$

It is important to note that $D = o(N)$. Taking the limit of N going to infinity, we get

$$T \geq eD\ln(N) \tag{34}$$

We need to modify (32) to get a tail bound on T, since we must show that the number of tests T

decays to zero as $N^{-\delta}$. Hence,

$$Tg\left(\frac{N-d}{N}\right)^{g} \geq \chi(N-d)\text{ln}(N-d) \tag{35}$$

Thus, (35) corresponds to the event that all the non-defective items have not been collected if $\chi(N-d)\text{ln}(N-d)$ total items have been collected. As stated previously, the probability of this event is at most $(N-d)^{-\chi+1}$.

By the Chernoff bound, the probability that the actual number of items in the negative tests is smaller than the expected number times (1 - ρ) is at most exp $\left(-\rho^2 T\left(\frac{N-d}{N}\right)^{g}\right)$. Hence, we have the event

$$(1-\rho)Tg\left(\frac{N-d}{N}\right) \geq \chi(N-d)\text{ln}(N-d) \tag{36}$$

Consider the union bound over both events (35) and (36). The probability that (36) does not hold is at most

$$exp\left(-\rho^2 T\left(\frac{N-d}{N}\right)^{g}\right) + (N-d)^{-\chi+1} \tag{37}$$

Optimizing for g, we substitute the optimal value obtained before i.e. $g^* = 1/\text{ln}\frac{N}{N-d}$. Note that

$$g^* = 1/\text{ln}\frac{N}{N-d} \tag{38}$$

$$e^{-g^*} = \frac{N-d}{N} \tag{39}$$

$$\left(e^{-1/g}\right)^{g} = e^{-1} \tag{40}$$

Therefore, for large N,

$$T \geq \frac{\chi}{1-\rho}\frac{(N-d)\text{ln}(N-d)}{g^* 1/\text{ln}\frac{N}{N-d}} \tag{41}$$

$$\approx \frac{\chi}{1-\rho}\frac{(N-d)\text{ln}(N-d)}{e^{-1}\frac{1}{\text{ln}\left(\frac{N}{N-d}\right)}} \tag{42}$$

$$= \frac{\chi}{1-\rho}\frac{(N-d)\text{ln}(N-d)\text{ln}\left(\frac{N}{N-d}\right)}{e^{-1}} \tag{43}$$

Using ln(1 + x)$\geq x - x^2/2$ with x = $D/(N-D)$, the above expression simplifies to

$$T \geq \frac{\chi}{1-\rho}e\left(D - \frac{D^2}{2(N-d)}\right)\text{ln}(N-d) \tag{44}$$

By choosing T greater than the bound in (44), we can further reduce probability of error. Hence, choose

$$T \geq \frac{\chi}{1-\rho} eD\text{ln}(N-d) \tag{45}$$

This will still imply a probability of error at most as large as (37).

By choosing $\rho = \frac{1}{2}$ and noting that $D \geq d$, we substitute (45) into (37). This implies that probability of error P_e satisfies

$$P_e \leq e^{-\frac{\rho^2\chi}{1-\rho}d\text{ln}(N-d)} + (N-d)^{-\chi+1} \tag{46}$$

$$= (N-d)^{-\frac{\rho^2\chi}{1-\rho}d} + (N-d)^{-\chi+1} \tag{47}$$

$$\leq 2(N-d)^{-\chi+1} \tag{48}$$

Taking $2(N-d)^{-\chi+1} = N^{-\delta}$, we have

$$\chi = \delta\frac{\text{log}N}{\text{log}(N-d)} + \frac{1}{\text{log}(N-d)} + 1 \tag{49}$$

For large N, χ approaches $\delta + 1$.

Therefore, we see that for sufficiently large N and probability of error at most $N^{-\delta}$, the following number of tests suffice to satisfy the probability of error condition as stated in the theorem.

$$T \geq 2(1+\delta)eD\text{ln}N \tag{50}$$

Thus, the proof is complete.

2) Column Matching Algorithm: In this proof, we have two cases :

- **Case 1** (Noiseless)- Errors: only false positives

  Specified upper bound :

$$T \leq eD(1+\delta)\text{ln}(n) \tag{51}$$

- **Case 2** (Noisy) - Errors : false positives and/or false negatives

  Specified upper bound :

$$\frac{16(1+\sqrt{\gamma})^2(1+\delta)\text{ln}2}{(1-e^{-2})(1-2q)^2}D\text{log}n \tag{52}$$

Here, $\gamma = \dfrac{\Gamma+\delta}{1+\delta}$, $\Gamma = \frac{ln(D)}{ln(n)} \in [0,1)$, D is the upper bound on number of defective items and q is the probability that the test result differs from the true result.

Let us consider the first case.

Case 1

False positives occur when column correponding to non-defectives is “hidden” by columns correponding to defectives. Let j be the column corresponding to a non-defective item and $j_1, j_2, ...j_d$ be columns corresponding to defective items. Then, probability that $a_{ij} = 1$ and at least one of $a_{ij_1}, a_{ij_2}..a_{ij_d}$ = 1 is $p(1-(1-p)^d)$. Hence, probability that the j^{th} column is hidden is $(1-p(1-p)^d)^T$.

Taking the union bound, probability of false positives in the noiseless case is :

$$P_e^+ \leq (n-d)(1-p(1-p)^d)^T \tag{53}$$

Optimizing w.r.t p, we get the optimal value of $p = 1/d$ or $1/D$. Set $T = \beta D\text{ln}(n)$. Then,

$$\begin{aligned} P_e^+ &\leq (n-d)(1-\frac{1}{D}(1-\frac{1}{D})^d)^{\beta D\text{ln}(n)} \\ &\leq (n-d)(1-\frac{1}{De})^{\beta D\text{ln}(n)} \qquad ((1-\tfrac{1}{x})^x \geq e^{-1}) \end{aligned}$$

Choose $\beta = (1+\delta)e$ to ensure the required decay. Hence

$$T \leq (1+\delta)eD\text{ln}(n) \tag{54}$$

Thus, the case 1 proof is complete.

Case 2

In the presence of noise, false positives and/or false negatives are possible. Hence, the total probability of error will be the sum of both these probabilities.

$$P_e = P_e^- + P_e^+$$

False negatives occur if more than expected number of ones get flipped to zeros in $\mathbf{y}$ in locations corresponding to ones in defective column. The probability of false negatives is given as :

$$P_e^- = \bigcup_{i=1}^{d} P(|\mathcal{T}_i| = t)P(|\mathcal{S}_i| < |\mathcal{T}_i|\,(1 - q(1+\tau))) \tag{55}$$

$$\leq d\left[\sum_{t=0}^{T}\binom{T}{t}p^t(1-p)^{T-t}\right]\left[\sum_{r=t-t(1-q(1+\tau))}^{t}\binom{t}{r}q^r(1-q)^{t-r}\right] \tag{56}$$

$$\leq d\,\exp\left[-\beta\log n(1-e^{-2})(q\tau)^2\right] \tag{57}$$

Here, $\mathcal{T}_i$ is the indicator set, specifying the number of ones in the column corresponding to the defective item. $\mathcal{S}_i$ is the matching set giving the indices of the matching entries of the column and the vector $\mathbf{y}$. The expression for P_e^- is a union bound taken over all defectives over all possible number of ones in the column (the first term) and the probability that the number of matches is less than the threshold (second term).

Since $P_e^- \leq n^{-\delta}$, β^- satisfies :

$$\beta^- > \frac{\left(\frac{\ln d}{\ln n} + \delta\right)\ln 2}{(1-e^{-2})(q\tau)^2} \tag{58}$$

False positives may occur if 1's in column are hidden or masked due to noise. Hence, we define f as the probability that a 1 in the location a_{ij} is hidden by ones in other columns or hidden by noise.

$$f = 1 - [(1-q)(1-p)^d + q(1-(1-p)^d)] \tag{59}$$

$$= 1 - q - \left(1 - \frac{1}{D}\right)^d (1-2q) \tag{60}$$

Let $D \geq 2$ and recall that $d \leq D$. ($D = 1$ is treated as a special case, with $p = 1/(D+1)$). Upper bound on f is :

$$\max_{D\geq 2, d\leq D} f = \max_{D\geq 2, d\leq D}\left(1 - q - \left(1-\frac{1}{D}\right)^d(1-2q)\right)$$

$$= (1-q) - (1-2q)/4$$

Probability of false positives is thus :

$$\begin{aligned}
P_e^+ &= \bigcup_{i=1}^{n-d} P(|\mathcal{T}_i| = t)P(|\mathcal{S}_i| \geq |\mathcal{T}_i|\,(1 - q(1+\tau))) \\
&\leq \left[(n-d)\sum_{t=0}^{T}\left(\begin{array}{c} T \\ t \end{array}\right)p^t(1-p)^{T-t}\right]\left[\sum_{r=t(1-q(1+\tau))}^{t}\left(\begin{array}{c} t \\ r \end{array}\right)f^r(1-f)^{t-r}\right] \\
&\leq (n-d)\exp(-\beta\log n(1-e^{-2})(1-2q)/(4-q\tau)^2)
\end{aligned}$$

Now $P_e^+ = n^{-\delta}$ gives us this expression :

$$\beta^+ > \frac{\left(\frac{\ln(n-d)}{\ln n} + \delta\right)\ln 2}{(1-e^{-2})((1-2q)/4 - q\tau)^2}$$

We see that during the derivation of the above expression, $\tau \in (0, (1-2q)/4q)$. The lower limit arises from the fact that a low τ will lead to a high probability of false negatives. Hence, τ must be greater than zero for the probability of false negatives to be low. The upper limit stems from a condition required in order to apply the Chernoff bound and obtain the expression for β^+.

Note that β^- is strictly increasing, and β^+ is strictly decreasing in this region. Since β must be at least as large as $\max\{\beta^-, \beta^+\}$, a good choice for β is the intersection of these curves.

Define

$$\gamma = \lim_{n,d\to\infty} \frac{\ln d + \delta \ln n}{\ln(n-d) + \delta \ln n} \tag{61}$$

And

$$\Gamma = \lim_{n,d\to\infty} \frac{\ln d}{\ln n} \tag{62}$$

Hence

$$\gamma = \frac{\Gamma + \delta}{1 + \delta} \tag{63}$$

Equating expressions for β^- and β^+, the optimal τ^* satisfies

$$\frac{\ln 2}{(1-e^{-2})/((1-2q)/4 - q\tau^*)^2} = \frac{\gamma \ln 2}{(1-e^{-2})(q\tau^*)^2} \tag{64}$$

Simplifying, we get

$$\tau^* = \frac{1-2q}{4q(1+\gamma^{-1/2})} \tag{65}$$

From this, we get the explicit bound for large n :

$$\beta = \frac{16(1+\gamma^{-0.5})^2(\Gamma+\delta)\ln 2}{(1-e^{-2})(1-2q)} \tag{66}$$

Thus, the case 2 proof is complete.

3) LP Decoding Algorithm: Given upper bound for the LP Decoding Algorithm:

$$T \leq \beta_{LP} D \ln n$$

where $\beta_{LP} = \max\{\frac{4e(\delta+1+\Gamma)}{(1-2q)^2}, 8e(\delta+1+\Gamma), \frac{4e(1-q+2qe)(\delta+1+\Gamma)}{(1-q)^2}, \frac{8e(\delta+1+\Gamma)}{(1-q+2qe)}, \frac{(1-q+qe)(\delta+\Gamma)(1+e)^2}{e(1-2q)^2}, \frac{8e(\delta+\Gamma)}{(1-q+qe)}\}$

Here, $\Gamma = \frac{ln(D)}{ln(n)} \in [0,1)$, D is the upper bound on number of defective items and q is the probability that test result differs from true result.

The outline of the proof is as follows.

- Define sets $\mathbf{\Phi}', \mathbf{\Phi}''$ containing *perturbation vectors* which depend on $\mathbf{x}$.
- Claim 9 says that any feasible $\bar{\mathbf{x}}$ that satisfies the conditions of the LP can be written as a sum of $\mathbf{x}$ and a non-negative linear combination of perturbation vectors. The 'linear combination' characterizes the directions in which a vector can be perturbed in a "finite" manner (instead of having to consider the infinite number of possible directions). The 'non-negativity' ensures that the objective function of the LP can only increase when perturbed
- Claim 10, 11 say that expected change (over randomness in the matrix $\mathbf{A}$ and noise ν) in each slack variable η_i when $\mathbf{x}$ is perturbed is strictly positive with high probability.
- In Claim 13, using Chernoff bounds from Claim 12, we show that expected change in the value of the objective function (weighted sum of changes in slack variables) for each perturbation vector is also strictly positive with high probability.
- Finally, since the set of feasible $(\bar{\mathbf{x}}, \eta)$ forms a convex set, value of the objective function corresponding to $\bar{\mathbf{x}}$ is greater than value of the objective function corresponding to $\mathbf{x}$. Thus, the LP decodes correctly as $\bar{\mathbf{x}} = \mathbf{x}$.

Without loss of generality, assume that $\mathbf{x}$ has 1s in the first d locations and 0s in the last $n-d$ locations. Define two sets :

- $\mathbf{\Phi'} = \{\phi'\}_{k'=1}^{d(n-d)}$

  This vector has one -1 in supp($\mathbf{x}$), one 1 outside supp($\mathbf{x}$) and 0s elsewhere.
- $\mathbf{\Phi''} = \{\phi''\}_{k''=1}^{d}$

  This vector has one -1 in supp($\mathbf{x}$) and 0s elsewhere.

Claim 9 : Any vector $\bar{\mathbf{x}}$ that satisfies the constraints in NCBP-LP can be written as

$$\bar{\mathbf{x}} = \mathbf{x} + \sum_{k'=1}^{d(n-d)} c_{k'}\phi'_{k'} + \sum_{k''=1}^{d} c_{k''}\phi''_{k''}, c_{k'}, c_{k'\prime} \geq 0 \tag{67}$$

Proof:

First stage : Let $\mathbf{x}$ be n bottles, with the first d bottles full, and the remaining empty. $\bar{\mathbf{x}}$ is another state of these bottles with $\bar{d} \leq d$ liters. In the first stage, we throw away $d - \bar{d}$ liters, taking care not to undershoot.(This corresponds to the perturbation vectors in Φ'')

Second Stage : For each bottle j among the first d bottles that still has more water remaining than in the corresponding bottle in the final state, use its water to increase the water level of bottles among the last d bottles, taking care not to overshoot. (This corresponds to the perturbation vectors in Φ'). Thus, it is "Conservation of mass". ■

Cost Perturbation Random Variables are defined as follows :

$$\triangle'_{0,i} = \eta_i(\bar{\mathbf{x}}) - \eta_i(\mathbf{x}), \triangle'_{0,i}\prime = \eta_i(\bar{\mathbf{x}}) - \eta_i(\mathbf{x}),\ \text{on } \hat{y}_i = 0 \tag{68}$$

$$\triangle'_{1,i} = \eta_i(\bar{\mathbf{x}}) - \eta_i(\mathbf{x}), \triangle'_{1,i}\prime = \eta_i(\bar{\mathbf{x}}) - \eta_i(\mathbf{x}),\ \text{on } \hat{y}_i = 1 \tag{69}$$

Claim 10 : The cost perturbation random variables take values only in $\{-1, 0, 1\}$.

Proof:

1) Case : $\hat{y}_i$ = 0 implies that $\eta_i(\mathbf{x}) = \mathbf{a_i}\mathbf{x}$. Hence,

$$\triangle''_{0,i} = \mathbf{a_i}(\mathbf{x''} - \mathbf{x}) = \mathbf{a_i}\phi''$$
$$\triangle'_{0,i} = \mathbf{a_i}(\mathbf{x'} - \mathbf{x}) = \mathbf{a_i}\phi'$$

   $\mathbf{a_i}$ = 0/1 vector and ϕ'' has exactly one non-zero component (equaling - 1), and ϕ' has exactly one component equaling - 1 and one equaling 1. Hence, the values of the cost perturbation variables are either -1, 0 or 1.

2) Case : $\hat{y}_i$ = 1 implies that the minimum value of $\eta_i(\mathbf{x})$ occurs at (1 - $\mathbf{a_i}\mathbf{x}$)$^+$(i.e the value is $(1 - \mathbf{a_i}\mathbf{x})$ if it is positive and 0 otherwise).

   Since $\mathbf{a_i}, \mathbf{x'}, \mathbf{x''}, \mathbf{x}$ are 0/1 vectors, $\eta_i(\mathbf{x'}), \eta_i(\mathbf{x''}), \eta_i(\mathbf{x})$ are 0/1 vectors. Hence, their pairwise differences have values in $\{-1, 0, 1\}$.

■

Claim 11

$$P(\triangle''_{1,i} = 1) = p(1-p)^{d-1}(1-q) \tag{70}$$

$$P(\triangle''_{1,i} = -1) = 0 \tag{71}$$

$$P(\triangle''_{0,i} = 1) = 0 \tag{72}$$

$$P(\triangle''_{0,i} = -1) = pq \tag{73}$$

$$P(\triangle'_{1,i} = 1) = p(1-p)^{d}(1-q) \tag{74}$$

$$P(\triangle'_{1,i} = -1) = p(1-p)^{d}q \tag{75}$$

$$P(\triangle'_{0,i} = 1) = p(1-p)[(1-q)(1-p)^{d-1} + q] \tag{76}$$

$$P(\triangle'_{0,i} = -1) = p(1-q)q \tag{77}$$

The proof proceeds by case analysis.

Proof:

Case Analysis:

- In (72) and (73), $\triangle''_{0,i} = \eta_i(\mathbf{x''}) - \eta_i(\mathbf{x}) = \mathbf{a_i}(\mathbf{x''} - \mathbf{x}) = \mathbf{a_i}\phi''$. Since ϕ'' only has one negative component and no positive components, $\triangle''_{0,i}$ is never positive. And $\triangle''_{0,i}$ = -1 when support of $\mathbf{a_i}$ intersects supp($\mathbf{x}$) at the location where $\phi'' = -1$. This has probability pq.
- In (70) and (71), for $\triangle''_{1,i}$, minimum values of $\eta_i(\mathbf{x}) = (1 - \mathbf{a_i}\mathbf{x})^{+}$. (i.e the value is $(1 - \mathbf{a_i}\mathbf{x})$ if it is positive and 0 otherwise.) This has values 0/1. Also,

  $$\text{supp}(\mathbf{x''}) \subset \text{supp}(\mathbf{x})$$

  with $\mathbf{x''}$ having exactly one less positive component. Hence

  $$\eta_i(\mathbf{x''}) \geq \eta_i(\mathbf{x})$$

  Hence $\triangle''_{1,i} \geq 0$. This is a strict inequality iff $\mathbf{a_i}\mathbf{x''} = 1, \mathbf{a_i}\mathbf{x} = 0$. This happens when support of $\mathbf{a_i}$ intersects that of $\mathbf{x}$ at one location (hence $y_i = 1$) and $\phi'' = -1$ in this location. The probability of this even is as given in Claim 11.

Similar analysis for the remaining two cases proves Claim 11.

■

Claim 12: Chernoff bound (multiplicative form) Let $\{W_i\}_{i=1}^{T}$ be a sequence i.i.d. binary random variables with probability distribution P(W_i = 1) = θ.

$$P(\left|\sum_{i=1}^{T}\frac{W_i}{T}-\theta\right|>\sigma)\leq \exp(-2T\sigma^2) \qquad \text{(additive form)} \tag{78}$$

$$P(\left|\sum_{i=1}^{T}\frac{W_i}{T}-\theta\right|>\sigma\theta)\leq \exp(-\frac{T\sigma^2\theta}{2}) \qquad \text{(multiplicative form)} \tag{79}$$

Finally, Claim 13 shows that regardless the direction of perturbation of $\mathbf{x}$, as long as it remains within the feasible set for the LP, the value of the objective function of the LP increases. This implies that the LP performs correct minimization.

Claim 13 : Choose T as $\beta_{LP}D\log n$. Then the LP fails with a probability $n^{-\delta}$.

Proof :

Each $\mathbf{a_i}$ chosen independently. This implies that the cost perturbation random variables are i.i.d.

- **Step 1** Define sets $\mathcal{S}_1' = \{i \mid \triangle_{1,i}' \neq 0\}, \mathcal{S}_0' = \{i \mid \triangle_{0,i}' \neq 0\}, \mathcal{S}'' = \{i \mid \triangle_{1,i}'', \triangle_{0,i}'' \neq 0\}$
- **Step 2** Calculate expected sizes of sets, (from Claim 11 and by substituting T, $p = 1/D$) :

$$E(|S_1'|) = Tp(1-p)^d > \frac{\beta_{LP}\log n}{e} \tag{80}$$

$$E(|S_0'|) = Tp(1-p)((1-q)(1-p)^{d-1}+2q) > \frac{\beta_{LP}(1-q+2qe)\log n}{e} \tag{81}$$

$$E(|S''|) = Tp((1-p)^{d-1}(1-q)+q) > \frac{\beta_{LP}(1-q+qe)\log n}{e} \tag{82}$$

- **Step 3** Use concentration inequalities (Claim 12). For some positive constants $\sigma_1', \sigma_0', \sigma''$,

$$P(\left|\mathcal{S}_1'\right| < E(\left|\mathcal{S}_1'\right|)(1-\sigma_1')) < n^{-\dfrac{\beta_{LP}(\sigma_1')^2}{2e}} \tag{83}$$

$$P(\left|\mathcal{S}_0'\right| < E(\left|\mathcal{S}_0'\right|)(1-\sigma_0')) < n^{-\beta_{LP}\left(\dfrac{1-q+2qe}{2e}\right)(\sigma_0')^2} \tag{84}$$

$$P(\left|\mathcal{S}''\right| < E(\left|\mathcal{S}''\right|)(1-\sigma'')) < n^{-\beta_{LP}\left(\dfrac{1-q+qe}{2e}\right)(\sigma'')^2} \tag{85}$$

- **Step 4** Calculating conditional probabilities and taking the limit $D \to \infty$

$$P(\triangle_{1,i}' = 1 \mid i \in \mathcal{S}_1') = 1-q \qquad P(\triangle_{1,i}' = -1 \mid i \in \mathcal{S}_1') = q \tag{86}$$

$$P(\triangle_{1,i}' = 1 \mid i \in \mathcal{S}_1') = \frac{(1-p)^{d-1}(1-q)+q}{(1-p)^{d-1}(1-q)+2q} \qquad P(\triangle_{1,i}' = -1 \mid i \in \mathcal{S}_1') = \frac{q}{(1-p)^{d-1}(1-q)+2q} \tag{87}$$

$$P(\triangle_{1,i}'' = 1 \mid i \in \mathcal{S}'') = \frac{(1-p)^{d-1}(1-q)}{(1-p)^{d-1}(1-q)+q} \qquad P(\triangle_{0,i}'' = -1 \mid i \in \mathcal{S}'') = \frac{q}{(1-p)^{d-1}(1-q)+q} \tag{88}$$

- **Step 5** Find objective value perturbation
  - For $\mathbf{x}' = \mathbf{x} + \phi'$, objective value perturbation :

$$\sum_{i \in \mathcal{S}_1'} (1(\triangle_{1,i}' = 1) - 1(\triangle_{1,i}' = -1)) + \frac{1}{e} \sum_{i \in \mathcal{S}_0'} (1(\triangle_{0,i}' = 1) - 1(\triangle_{0,i}' = -1)) \tag{89}$$

  - For $\mathbf{x}'' = \mathbf{x} + \phi''$, objective value perturbation :

$$\sum_{i \in \mathcal{S}''} (1(\triangle_{1,i}'' = 1) - 1(\triangle_{1,i}'' = -1)) + \frac{1}{e} \sum_{i \in \mathcal{S}''} (1(\triangle_{0,i}'' = 1) - 1(\triangle_{0,i}'' = -1)) \tag{90}$$

- **Step 6** Show that objective value perturbation is positive with high probability.

In (89), the first term is non-positive iff $\sum_{i \in \mathcal{S}_1'} (\triangle_{1,i}' = 1)$ equals $\sum_{i \in \mathcal{S}_1'} (\triangle_{1,i}' = -1)$. By concentration inequality (78) and (86), the probability of this event is at most $\exp\left(-\dfrac{|\mathcal{S}_1'|\,(1-2q)^2}{2}\right)$. But by (83) and (80), probability that $|\mathcal{S}_1'|$ is less than $\frac{\beta_{LP} \log N(1-\sigma_1')}{e}$ is at most $n^{-\frac{\beta_{LP}(\sigma_1')^2}{2e}}$. Then, probability that the first term is non-positive is :

$$n^{-\frac{\beta_{LP}(1-\sigma_1')(1-q)^2}{2e}} + n^{-\frac{\beta_{LP}(\sigma_1')^2}{2e}} \tag{91}$$

In (89), the second term is non-positive iff $\sum_{i \in \mathcal{S}_0'} (\triangle_{0,i}' = 1)$ equals $\sum_{i \in \mathcal{S}_0'} (\triangle_{0,i}' = -1)$. By concentration inequality (78) and (87), the probability of this event is at most $\exp\left(-\dfrac{|\mathcal{S}_0'|\,(1-q)^2}{2(1-q+2qe)^2}\right)$. But by (84) and (81), probability that $|\mathcal{S}_0'|$ is less than $\frac{\beta_{LP}(1-q+2qe) \log N(1-\sigma_0')}{e}$ is at most $n^{-\frac{\beta_{LP}(1-q+2qe)(\sigma_0')^2}{2e}}$. Then, probability that the second term is non-positive is :

$$n^{-\frac{\beta_{LP}(1-\sigma_0')(1-2q)^2}{2(1-q+2qe)e}} + n^{-\frac{\beta_{LP}(1-q+2qe)(\sigma_0')^2}{2e}} \tag{92}$$

In (90), note that $(\triangle''_{1,i} = -1))$ and $(\triangle''_{0,i} = 1)$ are always zero. Hence (90) is non-negative iff $\sum_{i\in\mathcal{S}''}(1(\triangle''_{1,i} = 1)$ equals $\sum_{i\in\mathcal{S}''} e(\triangle''_{0,i} = -1)$. By concentration inequality (78) and (88), this happens with a probability of at most $\exp\left(-\frac{|\mathcal{S}''|\, 2(1-2q)^2e^2}{(1-q+qe)^2(1+e)^2}\right)$. By (82) and (85), probability that $|\mathcal{S}''|$ is less than $\frac{\beta_L P(1-q+qe)\log(N)(1-\sigma'')}{e}$ is at most $n^{-\frac{\beta_L P(1-q+qe)(\sigma'')^2}{2e}}$. Hence the probability that (90) is non-positive is at most :

$$n^{-\frac{2\beta_{LP}e(1-\sigma'')(1-2q)^2}{(1-q+qe)(1+e)^2}} + n^{-\frac{\beta_{LP}(1-q+qe)(\sigma'')^2}{2e}} \tag{93}$$

- **Step 7** Now, using union bounds, the probability that any vector from $\mathbf{\Phi}' \cup \mathbf{\Phi}''$ causes a non-positive perturbation in the optimal values is

$$\begin{aligned} &d(n-d)\left(n^{-\frac{\beta_{LP}(1-\sigma_1')(1-q)^2}{2e}} + n^{-\frac{\beta_{LP}(\sigma_1')^2}{2e}} + n^{-\frac{\beta_{LP}(1-\sigma_0')(1-2q)^2}{2(1-q+2qe)e}} + n^{-\frac{\beta_{LP}(1-q+2qe)(\sigma_0')^2}{2e}}\right) + \\ &d\left(n^{-\frac{2\beta_{LP}e(1-\sigma'')(1-2q)^2}{(1-q+qe)(1+e)^2}} + n^{-\frac{\beta_{LP}(1-q+qe)(\sigma'')^2}{2e}}\right) \end{aligned} \tag{*}$$

  Choose $\sigma_0', \sigma_1', \sigma''$ to that $(*)$ is as small as possible. Here, the values is $\frac{1}{2}$ for each.
  Now choose β_{LP} as max $\left\{\frac{4e(\delta+1+\Gamma)}{(1-2q)^2}, 8e(\delta+1+\Gamma), \frac{4e(1-q+2qe)(\delta+1+\Gamma)}{(1-q)^2}, \frac{8e(\delta+1+\Gamma)}{(1-q+2qe)}, \frac{(1-q+qe)(\delta+\Gamma)(1+e)^2}{e(1-2q)^2}, \frac{8e(\delta+\Gamma)}{(1-q+qe)}\right\}$ so that maximum of the six terms in $(*)$ is less than $n^{-\delta}$. Hence, probability of strictly positive change in optimal value is (1 - $n^{-\delta}$). This is a high probability.
- **Step 8** Finally, the set of feasible $(\bar{\mathbf{x}}, \eta)$ is a convex set. Hence, if η strictly increases in any direction in $\mathbf{\Phi}', \mathbf{\Phi}''$, then it strictly increases for any perturbation direction.

Hence, true $\mathbf{x}$ must be the solution to the LP. Thus, the proof is complete.

REFERENCES

[1] C. L. Chan, S. Jaggi, V. Saligrama, S. Agnihotri *Non-Adaptive Group Testing: Explicit Bounds and Algorithms* ISIT 2012.

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

[3] D. Donoho *Compressed sensing* IEEE Trans. Inform. Theory, vol. 52, no. 4, pp. 1289–1306, Apr. 2006.

[4] Emmanuel Candès and Michael Wakin *An Introduction to Compressive Sampling.* IEEE Signal Processing Magazine, 25(2), pp. 21 - 30, March 2008.

[5] Richard Baraniuk, Justin Romberg, and Michael Wakin *Tutorial on Compressive Sensing* 2008 Information Theory and Applications Workshop.

[6] Mark Davenport, Marco Duarte, Yonina Eldar, and Gitta Kutyniok *Introduction to Compressed sensing*, Chapter in Compressed Sensing: Theory and Applications, Cambridge University Press, 2012.

[7] Holger Rauhut *Compressive Sensing and Structured Random Matrices*, Radon Series Comp. Appl. Math XX, 1–94

[8] J. A. Tropp and S. J. Wright *Computational methods for sparse solution of linear inverse problems* Proc. IEEE, invited paper, special issue, "Applications of sparse representation and compressive sensing," Vol. 98, num. 5, pp. 948-958, June 2010.

[9] Richard Baraniuk *Compressive sensing*. IEEE Signal Processing Magazine, 24(4), pp. 118-121, July 2007.

[10] M. Mitzenmacher and E. Upfal *Probability and Computing: Randomized Algorithms and Probabilistic Analysis* Cambridge University Press, 2005.

[11] D. Needell and J. A. Tropp *CoSaMP: Iterative signal recovery from incomplete and inaccurate samples* Appl. Comput. Harmon. Anal., Vol. 26, pp. 301-321, 2008.

[12] D. Malioutov, M. Malyutov *Boolean compressed sensing: LP relaxation for group testing*, Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on, pp. 3305–3308

[13] Stephen Boyd and Lieven Vandenberghe, *Convex Optimization*, Cambridge University Press

[14] Richard Baraniuk, Mark Davenport, Ronald DeVore, and Michael Wakin, *A simple proof of the restricted isometry property for random matrices*, Constructive Approximation, 28(3), pp. 253-263, December 2008