Projeto de Graduação

DEPARTAMENTO DE ENGENHARIA ELÉTRICA

July 2, 2018

THE GROUP TESTING PROBLEM VIA SPARSE GRAPH CODES

Pedro Abdalla Teixeira

Projeto de Graduação

DEPARTAMENTO DE ENGENHARIA ELÉTRICA

THE GROUP TESTING PROBLEM VIA SPARSE GRAPH CODES

Student: Pedro Abdalla Teixeira

Advisor: Marco Antônio Grivet Mattoso Maia

Acknowledgements

I would like to thank God, family, friends, in special:

- My advisor Prof. Marco Antônio Grivet to believe in me and in this project
- Prof Ramtin Pedarsani for presenting me the wonderful group testing problem
- To all my friends from eletrical engineering, mathematics and life. It is impossible to mention everyone and to exclude someone is equally impossible
- My uncle Elias, my aunt Clarice and my grandmother for the example of courage and determination, they will always be in my memory
- To my dear father, for the most important advice, let the book be your best teacher
- To all my brothers and sisters, in special, Ivan for being an example of professional dedication
- To my lovely mother, for the support, patience and example of mother. She will always be my number one.

Abstract

Data Science is one of the most active fields of research, over past decades, the number of mathematicians, statisticians, computer scientists and engineers working on the different subareas of data science has grown tremendously. Since we live in a digital era, our devices are storing and processing data all the time, then fast algorithms are useful to save time, money and energy consumption.

Group Testing problem arises in numerous areas of technology such as machine learning, coding theory, medicine and signal processing due to its low complexity. The main goal is to recover a set of K defective items from a total set of n items by performing the minimum quantity of tests as possible. Each test is related to a chosen subset of n. The test returns positive if there is at least one defective item in the subset tested and negative otherwise.

In this work we present the algorithm SAFFRON (**S**parse Gr**A**ph codes **F**ramework for G**RO**up Testi**N**g) proposed by the authors Kangwook Lee, Ramtin Pedarsani and Kannan Ranchamdran [1]. We discuss the sophisticated mathematics used, generalizations, we present the theoretical guarantees of the algorithm and we also support our work with numerical simulations as well.

Keywords: Random Sparse Graphs, Combinatorial Optimization, Group Testing, Data Science, Digital Communication

DEPARTAMENTO DE ENGENHARIA ELÉTRICA

O PROBLEMA DE TESTE DE GRUPO VIA GRAFOS SPARSOS EM CÓDIGOS

Resumo

Ciência dos dados é um dos ramos mais ativos em pesquisa, nas últimas decadas, o número de matemáticos, cientistas da computação, estatísticos e engenheiros atuando nesta área cresceu tremendamente. A partir do momento em que nos inserimos no mundo digital, nossos dispositivos eletrônicos estão armazenando e processando dados todo o tempo, então algoritmos de rápida eficiência são muito úteis para poupar tempo, recursos financeiros e até mesmo consumo de energia.

O problema chamado de teste de grupo aparece em inúmeras áreas como aprendizado de máquina, teoria dos códigos, medicina e processamento de sinais, devido à sua capacidade de atingir níveis de baixa complexidade computacional. O objetivo deste problema é identificar uma população de K itens nomeados de defeituosos dentre uma população total de n, através de testes que retornam positivo caso haja pelo menos um defeituoso no grupo testado e retorna negativo caso ao contrário.

Neste trabalho apresentamos o algoritmo SAFFRON (**S**parse Gr**A**ph codes **F**ramework for G**RO**up Testi**N**g) proposto pelos autores Kangwook Lee, Ramtin Pedarsani and Kannan Ranchamdran [1]. Discutimos a matemática sofisticada utilizada, generalizações, além disso apresentamos garantias de funcionamento através de demonstrações e por fim damos suporte ao trabalho com simulações númericas.

Palavras-chave: Grafos Aleatórios Esparsos, Otimização Combinatória, Teste de Grupo, Comunicações Digitais, Ciência dos Dados

DEPARTAMENTO
DE ENGENHARIA
ELÉTRICA

Summary

List of Figures

List of Tables

1 Introduction

a The History of Group Testing

Group Testing problem arose in the second world war [2], when the United States Public Health Service and Selective Service started a project to identify soldiers who was contaminated by the virus syphilis. To test if the patient was sick or not was too expensive, the only way to perform less tests was to test a group together, if the test returned positive, the doctors only knew that at least one among the group tested was sick, on the other hand if the test returned negative, they could discard this possibility and send the soldiers back to the war. The natural question that arose is related with the minimum number of tests required and the best strategy to identify the sick population, it would be a great advance in terms of money and time if the doctors could perform less tests.

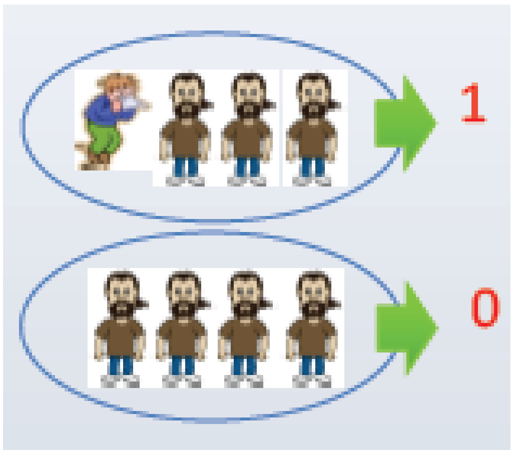

Figure 1: **Group Testing illustration**

In the beginning, group testing problem had nothing to do with the digital world, but nowadays, it is a very important tool to provide low complexity schemes. It has extensively studied in the literature in the past decades, in order to have an idea of how popular this problem becomes, researchers applied group testing algorithms in the following areas of human knowledge:

Spanning Across Biology: In the field of spanning across biology, a problem called DNA library screening relies on the challenge to "determine which clone (a DNA segment) from the library contains which probe from a given collection of probes [3]. The strategy to tackle this problem was to perform tests that returns positive if and only if the clone contains a probe.

Medicine: According to the authors in [4], the problem of pathogen identification involves the identification of disease-causing biomolecules. Drug discovery is the field dedicated to study methods to identify chemical compounds, called lead compounds, that bind to pathogenic proteins and eventually inhibit the function of the protein. The lead compounds are abstracted as inhibitors, pathogenic proteins as defectives, and the mixture of “ineffective” chemical compounds and non-pathogenic proteins as normal items. A defective could be immune to the presence of an inhibitor in a test. Then, a test containing a defective is positive if it does not contain its “associated” inhibitor.

Computer forensics: Computer forensics is a field that deals with methods to identify digital evidence of crime after suspect that some digital crime has been committed, for an example, someone trying to change information in a important document. According to the authors in [5] "a cryptography one-way hash is a commonly used way of detecting unauthorized or otherwise malicious modification of a file or other digital object. This is done by storing a keyed cryptography hash of the item and using it later as a reference for comparison. The problem seek to store as few hashes as possible so as to enable the pinpointing of which of these n items were modified ".

Dead Sensor Diagnosis: In a sensor network with a total number of n sensors, there are d sensors dead. The sensors can communicate with a base station using broadcast and respond protocol, the problem occurs when the sensor fail, it become invisible to the network. The solution to the problem is to send a group controller or a group test to figure out which sensors are "dead". The problem was formulated as group testing problem in [6]

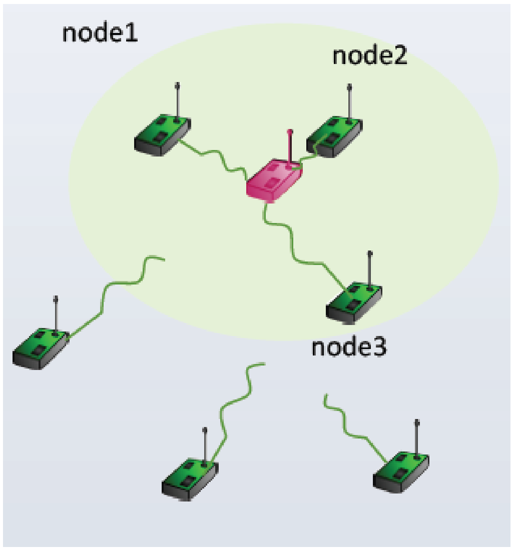

Figure 2: **Sensor Diagnosis Illustration**

As we mentioned all problems above can be formulated as a group testing problem, the list of applications is not exhaustive, there are other applications such as signal processing [7] and data analysis [8]. We now describe in more details another two applications, machine learning via Boolean compressed sensing and multi channel access. The reason for a deep description of these applications is due to these problems are very related with electrical engineering, in special, multi access channel is a key technology for digital communications.

b Machine Learning via Boolean Compressed Sensing

Machine learning is a field that lies in the intersection of artificial intelligence, computer and electrical engineering. Machine learning establish a new paradigm, the problem of learning, the computer "learn" data without being explicitly programmed. Some learning problems are related to sparse data recovery and intersects some fields such as compressed sensing, a field that studies problems related to how acquire data. For the readers not familiar with sparse recovery formulation see the appendix b

Briefly speaking, compressed sensing aims to recover sparse data from the observed vector in a incomplete structure. Formally, the problem is to find a sparse solution $\mathbf{x}$ from a linear system of equations $\mathbf{y} = \mathbf{A}\mathbf{x}$, where $\mathbf{A}$ is a matrix that models the system and has low rank. The popular approach to solve this problem is the following combinatorial optimization problem:

$$\begin{array}{rl} \min & ||x||_0 \\ \text{subject to} & \mathbf{y} = \mathbf{A}\mathbf{x} \end{array}$$

Where $||x||_0$ is the quasi-norm l^0 that counts the number of non zero entries. Clearly minimize the l^0 norm is a strategy to find sparse solutions, i.e, vectors with a few non-zero entries. The boolean compressed sensing is the same problem above, however since we are dealing with the digital world we do not have the common algebraic operations anymore. The problem is adapted as follows:

$$\begin{aligned} \min \quad & ||x||_0 \\ \text{subject to} \quad & \mathbf{y} = \mathbf{A} \vee \mathbf{x} \end{aligned}$$

Both optimization problems are NP-Hard [9]. The classical approach to relax these problems is the l^1 norm relaxation. For the first optimization problem is just replace the l^0 quasi-norm by the l^1 norm, however for the boolean case is much more complicated.

One of the major problems in machine learning is the problem called rule learning that deals with learning rules to describe patterns and regularities in a dataset [10]. In the celebrated work [11] the authors modeled the vector $\mathbf{y}$ to be the training vector with the system rule $\mathbf{A}$, then the vector $\mathbf{x}$ is the vector to be learned. They also proposed a linear program relaxation for the boolean case(second) as follows:

$$\begin{aligned} \min \quad & \sum_{i=1}^{n} x_i \\ \text{subject to} \quad & \mathbf{A}_P x \geq 1 \\ & \mathbf{A}_N x = 0 \\ & \forall i \in \{1, ..., n\} \; 0 \leq x_i \leq 1 \end{aligned}$$

Where $\mathbf{A}_P$ is formed by the rows corresponding to the $y_i = 1$, named as positive test part and $\mathbf{A}_N$ is formed by the others rows named as negative test part. In the section 3b we give more details about compressed sensing formulation for group testing.

c Multi Access Channels

Wireless communication is a key technology for digital communications schemes, multiple channels access problem occurs when a subset K of n devices wants to use the channel to transmit data. In this context, consider n devices sharing a communication channel, we want to know which device is transmitting data, however we have conflict when two or more devices are transmitting data at the same time.

The problem of multi access channel conflict is a little bit different from the others, in [6] the authors formulated the test as ternary test and not binary anymore. The test returns 0 with none of the devices tested are transmitting data, it returns 1 if only one of the devices are transmitting data and finally it returns a answer of conflict with two or more devices are transmitting data. We can view it as a group testing problem with ternary test named as multi level group testing problem, we want to identify a set of K transmitting ("defective") devices by performing ternary tests from a total of n devices connected to the channel.

We were motivated by the discovery that multilevel group testing could be applied for digital communications. For this reason we include in the chapter about generalizations of group testing a brief explanation on how multilevel is formally defined and how it is solved by a generalization of SAFFRON algorithm called MULTI-SAFFRON algorithm proposed in [12]. The following illustration 3 is due to [13]

d Notations

We include a table of all notations used along the text just to make the thesis more readable, for readers not familiar with the big $\mathcal{O}$ notation see the appendix a.

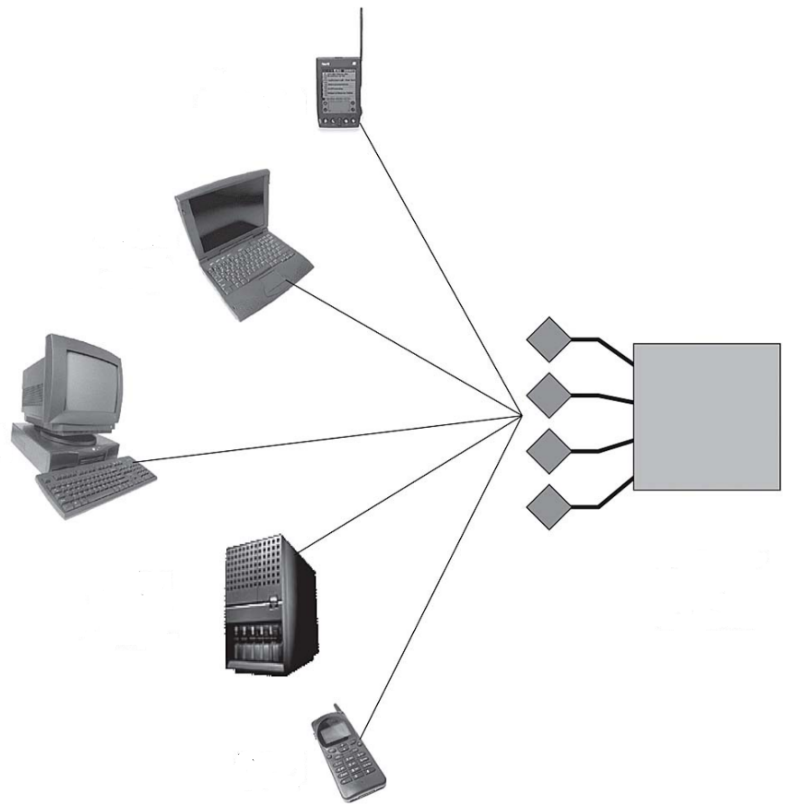

Figure 3: **Multi Access Channel**

Notations	Definition
n	Total number of items
K	Number of Defective items
m	Number of tests(pools)
M	Number of Right Nodes
$\mathbf{x}$	Binary representation of the set of defective items
$\mathbf{y}$	Binary representation of the group test results
$\mathbf{z}_i$	Right Node measurement from i-th node
$\mathbf{A}$	The group testing matrix
U	The signature matrix
u_i	The i-th column of the signature matrix
$\mathbf{b}_i$	The binary representation of the number i-1
$W(.)$	The Hamming weight of a code
$\bar{\mathbf{x}}$	The bit-wise complement of $\mathbf{x}$ or the bit-flipped vector of $\mathbf{x}$
$[n]$	$\{1, 2, ..., n\}$
$D_H(.,.)$	Hamming Distance between two codes
$\mathbf{T}$	The "adjacency matrix" of the bipartite graph
ϵ	Error Floor
d	Left node Degree
λ	Poisson distribution parameter
L	Number of Logical Levels
$\|\|.\|\|$	A norm of a vector
$\oplus$	Operation mod-2
$\vee$	Logical Operation "OR"
$\wedge$	Logical Operation "AND"

Table 1: **Table of Notations**

2 Theory of Random Sparse Graph Codes

a Graphs and Randomness

In this section we alternate intuition and formal definitions. To begin with, we formally define the notion of a graph. There several equivalent definitions, the definition below is based on [14] :

Definition 1. Given a set $V = [n]$ where $n \in \mathbb{N}$, a graph is a pair $G = (V, E)$ satisfying $E \subset [V]^2$

In other words, the elements of E are 2-elements of V. The elements of V are called nodes and elements of E edges, we say that a node i is connected to a node j if $\exists e \in E, e = \{i, j\}$.
Another important notion is the notion of a subgraph $H = (V(H), E(H))$ of the graph G.

Definition 2. A subgraph $H = (V(H), E(H))$ of G is a graph such that $V(H) \subset V(G)$ and $E(H) \subset E(G)$

For the sake of concreteness we give an example of the concept above, see figure 4.

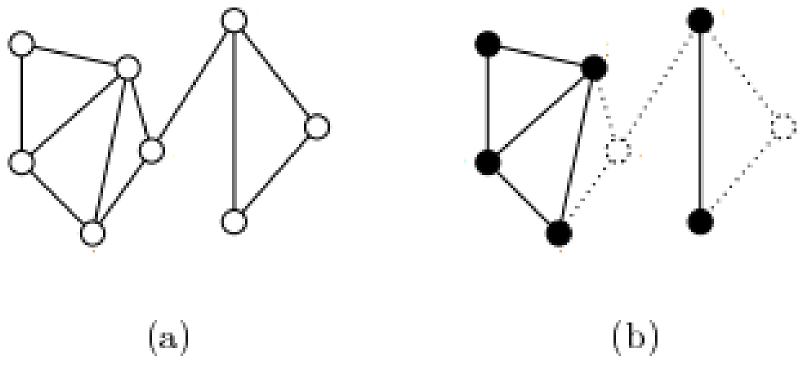

Figure 4: **Graph Illustrations**
Letter (a) provides an example of a graph and (b) provides an example of a subgraph in black

For practical purposes we restrict our attention to simple bipartite graphs. We say that a graph is said simple if none of vertices create a "loop", i.e, none of the vertices can connect with itself, formally :

Definition 3. A graph $G = (V, E)$ is said to be simple if $\forall v \in V, \{v, v\} \notin E$

Besides a graph is said bipartite if we can split it in two distinct parts and each edge has to connect a node in U to a node in W or vice-versa, we define in a formal way:

Definition 4. A graph $G = (V, E)$ is said to be bipartite if exists two sets U and W such that $U \cap V = \emptyset$, $U \cup W = V$ and $\forall \{v_1, v_2\} \in E \rightarrow (v_1 \in U \wedge v_2 \in W) \vee (v_1 \in W \wedge v_2 \in U)$

Finally a graph is said sparse if the number of edges has the same order as the number of nodes, we say that $|E| = O(|V|)$. The formal definition is analogous to the definition of a s-sparse vector in the appendix b :

Definition 5. A graph $G = (V, E)$ is said to be e-sparse if $|E| \leq e$

The following definitions will be useful in the study of decoding schemes.

Definition 6. For a simple graph $G = (V, E)$, the degree of a vertex $v \in V$ is the number of nodes connected with v. The set $N(v)$ is the set of nodes connected with v, named as neighbourhood of v

A important structure is the tree. We say that a graph G is connected if for every two nodes we have a path between them, i.e, a sequence of edges connecting the two nodes mentioned. A cycle is a path that begins and ends at the same node and finally a tree is a connected graph without cycles. Since the names are intuitive, we left to the readers to write the formal definitions or see in [14]. The figures 6 and 7 illustrates the definitions above:

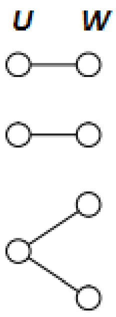

Figure 5: **Simple Bipartite Sparse graph illustration**

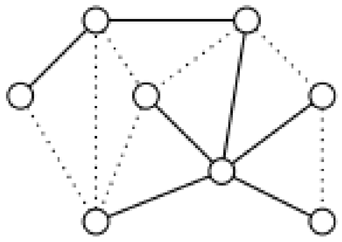

Figure 6: **A tree embedded in a graph**
The tree is represented by the lines in black only and not by the dashed lines

The pioneer work connecting probability theory and graph theory is due to Paul Erdos, who created the probabilistic methods that consists in nonconstructive methods to prove that certain graphs having some particular property exists. In the world of computer science, random graphs are used to provide an easier construction of many objects.

One of this objects is the random graph, it is a graph constructed in a probabilistic way. Consider the same set V as before and $\mathcal{G}(\mathcal{V})$ the set of all possible graphs created with V, we associate each possible edge e to a probability p_e, that represents the probability of an edge e belong to a graph chosen randomly from $\mathcal{G}(\mathcal{V})$. We denote the sample space by $\mathcal{G}(n, p)$ where n is the number of nodes and p the probability measure. For more formal constructions involving abstract sample spaces and sigma algebras see [14].

We can also consider stochastic process on graphs, for simplicity, lets consider only the Poisson graph that is defined to be a random graph following a Poisson distribution. It is well known fact from the probability theory that poisson distribution characterizes the arrival process.

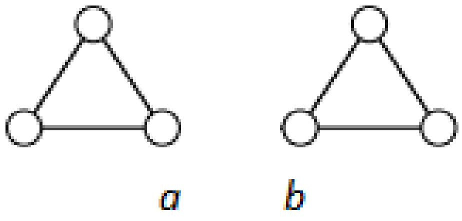

Figure 7: **A disconnected Graph**
Illustration of a graph that is not connected(disconnected), observe that there is no path between the nodes a and b

b Coding Theory

1 Basic concepts about Codes

Coding theory is a subfield of information theory concerned about recovering information corrupted by noise, sometimes the area is referred as error control coding. The main idea of this section is to introduce some ideas and theorems from modern coding theory that inspired and helped the authors to create SAFFRON. We focus only on the decoding procedure.

We set a binary code as a sequence of binary elements, elements of the field $\mathbb{F}_2$, equipped with operation mod-2, i.e, the logical operation XOR. We present the modern definition based on [15], for the readers that want to know about fields see the in the appendix definition 17. The notion of fields and more abstract algebra notions will not be required for the rest of this thesis.

Definition 7. A vector X is said to be a binary code of length N and cardinality M over a field $\mathbb{F}_2$ if it is a collection of M elements from $\mathbb{F}_2$, $X(N,M) = (x^1, ..., x^M)$, $x^m \in \{0,1\}$ $\forall 1 \leq m \leq M$

The elements of the code are called codewords. Now an example of the code that will be used in the next chapters:

Example 1. We always consider the length $N = 1$. Most of the cases $M = n$, in which n will be an arbitrary natural number, an example of code $X(1,4)$ is the vector $(1,0,0,1)$.

For a finite code X, the Hamming weight is defined to be the number of non-zero entries.

Definition 8. The Hamming weight of a code X, denoted by $W(X)$, is the norm $||X||_0$

The main importance of Hamming weight is to define the Hamming distance between two codes. Given two codes u_1 and u_2, the Hamming distance is given by XOR operator between these codes followed by the numerical sum.

Definition 9. Given two codes u_1 and u_2, the Hamming Distance between these codes, denoted by $D_H(u_1,u_2)$, is given by $D_H(u_1,u_2) = ||u_1 \oplus u_2||_0$

Example 2. Let $u_1 = (1,0,0,1)$ and $u_2 = (0,1,0,1)$ two codes. The Hamming weight and Hamming distance are given by:

$$W(u_1) = 1 + 0 + 0 + 1 = 2$$

$$W(u_2) = 0 + 1 + 0 + 1 = 2$$

$$D_H(u_1,u_2) = \sum_{i=1}^{4}(1,1,0,0) = 1 + 1 = 2$$

The Hamming distance is used to compute the bit error rate (BER). To compute the BER just compute the hamming distance and divide by the length of the vector.

2 SPA and Peeling Decoder

One of the most important decoding schemes based on sparse graph code principles was introduced by Robert Gallagher in his phd thesis, in which he introduced the Low Density Parity Check Codes (LDPC) [16]. The decoding scheme is based on a sparse bipartite graph called Tanner Graph, the graph is divided in two disjoint groups called variable nodes and check nodes and is represented by a matrix H, where $H_{ij} = 1$ if and only if the variable node i is connected with the check node j. This matrix is generated randomly, then its represents a sparse bipartite random graph. There are several methods to construct the parity check matrix, for example, a d-left regular bipartite construction, where each left node has a degree equal to d. Although this construction is not the best for LDPC codes, it will be useful in next chapters.

Example 3. Consider the following example of a parity check matrix H and its Tanner Graph, c represents check nodes, while v represents variable nodes

$$H = \begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

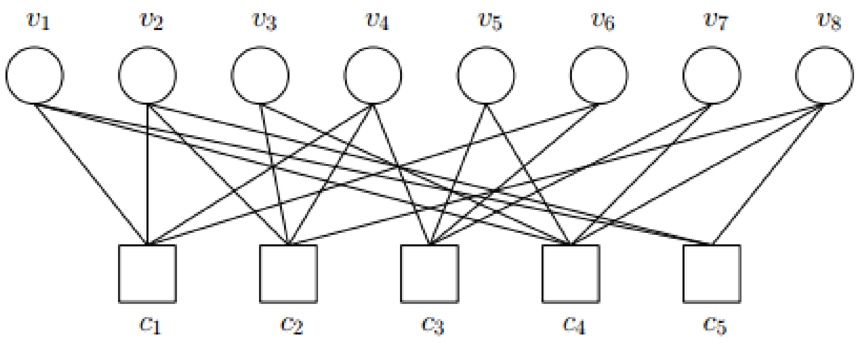

Figure 8: **Tanner Graph**

Message passing algorithms are algorithms that changes estimates about the reliability of the bit according to some statistical decision rule. One of the most famous is the sum product algorithm, a message passing algorithm used for decoding LPDC schemes. Our statistical decision rule is called MAP (Maximum A Posteriori Probability) and is given by the log likehood ratio:

$$L(v_j) = log\frac{P(v_j = 0|r)}{P(v_j = 1|r)} \tag{1}$$

In which r is the received information or prior knowlodge. We define the message passing rules, we denote $\mu^k_{c_i \to v_j}$ to be the message that is sent from a check node i to a variable node j at the iteration k and vice-versa.

$$\mu^k_{c_i \to v_j} = 2tanh^{-1}(\prod_{j' \in N(c_i)/j} tanh(\frac{\mu^k_{c_i \to v'_j}}{2})) \tag{2}$$

$$\mu^{k+1}_{v_j \to c_i} = L_{v_j} + \sum_{i' \in N(v_j)/i} \mu^k_{c'_i \to v_j} \tag{3}$$

$$L^{k+1}_{c_j} = L_{v_j} + \sum_{i \in N(v_j)} \mu^k_{c_i \to v_j} \tag{4}$$

Where $N(c_j)$ denotes the nodes connected with the check node j. Finally the SPA (Sum Product Algorithm) proceeds iterating the equations until a maximum number of iterations with initialization equal to zero for the check node information and the variable node information is the channel information that changes according to the channel. For more details about message passing algorithms we refer the readers [17] for a engineering point of view and [18] for a mathematical point of view

Sometimes the tanner graph is not sparse enough and take too much time to compute all the rules. In this context, peeling decoders arise to make this task easier, the basic principle is to remove the edges associated with low information updates, i.e, only keep the most useful information each step. Since SAFFRON is inspired in peeling decoding but does not apply directly the message rules mentioned we do not provide an example neither implementation of the algorithm. For more details about peeling decoders we refer the readers [15]

3 Density Evolution & Spatial Coupled Decoder

Density Evolution is a very complex tool to analyze message passing algorithms that made profound impacts on the field of information theory [15]. We restrict our attention to the fixed point characterization, the idea is to evaluate the probability of error p_j at iteration j, analyze the node as a tree-like neighbourhood, i.e, a node with a neighbourhood that does not contain and try to find some stability for the error, a fixed point.

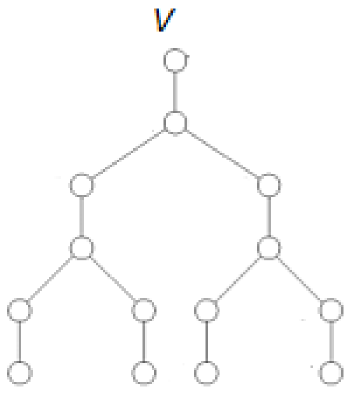

Figure 9: **Tree-Like Neighbourhood of v**

Formally we say that a real valued function $\gamma(p)$ is a contraction if $|\gamma(p_1) - \gamma(p_2)| \leq C|p_1 - p_2|$ where C is a constant between $(0, 1)$. The main theorem is the Banach fixed point theorem, we provide the version that we will apply later.

Theorem 1. *Let γ be a real valued function that is a contraction. The the sequence $x_{n+1} = \gamma(x_n)$ converges to a limit $p = \gamma(p)$. Moreover the set of fixed points $\{p : \gamma(p) = p\}$ contains a single point only.*

Proof. Observe that $|x_{n+1} - x_n| \leq |x_n - x_{n-1}|$, according to the D'Alembert criterion 10 for the series $s = \sum_{n\geq 1}(x_n - x_{n-1})$ is absolutely convergent. The sum of the first N terms is $S_N = x_N - x_0$, taking the limit as N goes to infinity we have $\lim x_n = s + x_0 = a$, the sequence converges, therefore exists a number a such that $\gamma(a) = a$. To end up with the proof, observe that if a and b are fixed points then $|a - b| \leq C|\gamma(a) - \gamma(b)|$ it follows directly that $C \geq 1$, its contradicts the assumption of γ be a contraction. $\square$

The mechanism of density evolution does not work if the nodes does not behave as a tree-like neighbourhood as the number of iterations goes to infinity, formally for each case we have to prove that the graph is a tree-like neighbourhood with high probability and take the limit the of iterations goes to infinity.

Density Evolution motivated new methods for the parity check matrix construction. Spatial Coupled decoder is a decoder created to improve performance of LDPC decoders, we are only going to use one theorem from the spatial coupled theory in the section 7. The main idea is to instead of applying a "big" matrix H that have the desired properties, spatial coupled schemes apply several "small" decoders in a sequence, the parity check matrix is given by $H = [H_1|...|H_n|$ where each H_i is a kind of LDPC decoder described above. Although the idea is simple, the analysis of this kind of LDPC are hard, see for more details [19].

3 Group Testing

a Classical Formulation

As we mentioned group testing tackles the problem to identify a set of K defective among a set of n items performing tests. In this sense there are some considerations about how the sequence of tests are made, if the sequence has to be done at once, i.e, you define all tests before the first set tested, we refer to this procedure as non-adaptive group testing. Formally, we define first the pools(subset tested) P_1, P_2, P_M, all are subsets of $[n]$, in which M is the total numbers of tests. The test is a function T that assumes 1 if $P_i \cap K \neq \emptyset$ and 0 otherwise. The goal is to obtain a set K' after performing all tests that is equal to K, we refer this problem as a combinatorial non-adaptive group testing

On the other hand if you can define the set tested after having the information of a previous test, we refer to it as a adaptive group testing. Formally you can define $P_i = f(P_1, ..., P_{i-1})$ where f is a function that related the next pool with the previous one. The definition of test and the goal of the problem still the same. This strategy is referred as combinatorial adaptive group testing problem.

Another important consideration is about the missclassifications, if we consider the group testing problem aiming a perfect recovered set, i.e, no matter the computational complexity, the recovered set has to be exactly equal to the set of defective items, we call as a combinatorial group testing problem. In many practical applications, in order to have lower complexity, we relax a bit the problem, we consider that a small number(less than a target ϵ) of missclassifications can occur with a very low probability, a probability that goes to zero as n goes to infinity. We refer to this kind of group testing as a probabilistic group testing problem.

Remark 1. In this work, we only study non adaptive schemes due to its advantage in architecture because we can make it parallelized, adaptive group testing is sequential in nature. We also consider the probabilistic approach due to the priority for communication schemes is the low computational complexity

To end up with this section we relate the readers with the previous results in non adaptive group testing problem. A well known lower bound for combinatorial group testing is $\Omega(\frac{K^2 logn}{logK})$ [20]. The best known algorithm for combinatorial non-adaptive case is [21] that performs $\mathcal{O}(K^2 logn)$ tests. For probabilistic group testing SAFFRON is the best known algorithm with order-primal in complexity $\mathcal{O}(Klog_2 n)$, before SAFFRON the best known was GROTESQUE proposed in [22] that performs $\mathcal{O}(KlogKlogn)$

b Compressed Sensing Formulation

Compressed sensing is a new paradigm to acquire data from linear measurements. The main goal is to recover a unknown vector $\mathbf{x}$ from the observed vector $\mathbf{y} = \mathbf{A}\mathbf{x}$ and the matrix $\mathbf{A}$ has low rank and it is called the sensing matrix. The vector $\mathbf{y}$ is modeled as a list of linear measurements represented by inner products $\langle , \rangle$. Let $\mathbf{a_i}$ be the i^{th} row of the matrix $\mathbf{A}$

$$\mathbf{y} = \mathbf{A}\mathbf{x} = \begin{bmatrix} \langle \mathbf{a}_1, \mathbf{x} \rangle \\ \vdots \\ \langle \mathbf{a}_m, \mathbf{x} \rangle \end{bmatrix}. \tag{5}$$

The key idea of SAFFRON is to model group testing problem using sparse signal recovery principle and sparse graph codes, so we apply the same idea as above in compressed sensing to design a sensing matrix $\mathbf{A}$ and decode the observations measured using a peeling-like decoder based on sparse graphs instead of solving directly the optimization problems introduced in the first chapter. Therefore our task is to recover a binary unknown vector $\mathbf{x} \in \{0,1\}^n$,of which $\mathbf{x_i} = 1$ if and only if item i is defective. The linear measurement will be our test, therefore we define the ith row to be $\mathbf{a_i} \in \{0,1\}^n$, of which $\mathbf{a_{ij}} = 1$ if and only if the item j belong to the subset tested(pool). Formally the test and the observed vector $y = (y_1, .., y_m)^T$ are given by:

Definition 10. We define the observation vector as $\mathbf{y}$ and its measurements(tests) to be y_i. They are given by :

$$\mathbf{y_i} = \langle \mathbf{a_i}, \mathbf{x} \rangle = \bigvee_{j=1}^{n} a_{ij} x_j \tag{6}$$

$$\mathbf{y} = \mathbf{A} \odot \mathbf{x} = \begin{bmatrix} \langle \mathbf{a}_1, \mathbf{x} \rangle \\ \vdots \\ \langle \mathbf{a}_m, \mathbf{x} \rangle \end{bmatrix}. \tag{7}$$

The logical operator "OR" between many bits is represented by the symbol $\bigvee$. Now we define the group testing matrix

Definition 11. The matrix $\mathbf{A}$ is called the group testing matrix $\mathbf{A} \equiv (a_1, ..., a_m)^T \in \{0,1\}^{m \times n}$

Our goal is to design a decoding function $g_{\mathbf{A}} : \{0,1\}^m \rightarrow \{0,1\}^n$ that decodes a vector equal to the target vector $\mathbf{x}$,i.e, $g_{\mathbf{A}}(y) = x$.

Consider now a bipartite graph with n left nodes representing the items and M right nodes representing the bundle of tests, each right node is a subset of the set of total items. The graph is designed at random with d regular constraint, each left node is connected to exactly d right nodes. We construct the matrix related to the graph as $\mathbf{T} \in \{0,1\}^{M \times n}$, where $\mathbf{T}_{ij} = 1$ if and only if the right node i is connected to the left node j, we name as adjacency matrix. The authors assign an arbitrary number, lets say h tests to every right node based on a signature matrix $\mathbf{U}$ as follows. Assume that $\mathbf{t}_i \in \{0,1\}^n$ denotes the ith row of $\mathbf{T}$. The sensing matrix of SAFFRON is associated to the ith right node with the equation $\mathbf{A}_i = \mathbf{U}\,\mathrm{diag}(\mathbf{t}_i) \in \{0,1\}^{h \times n}$ and the overall sensing matrix is $\mathbf{A} = [\mathbf{A}_1^T, \cdots, \mathbf{A}_M^T]^T$. Therefore, the total number of tests is $m = M \times h$. The signature matrix is given by :

Definition 12. The signature matrix of SAFFRON is defined to be :

$$\mathbf{U} = \begin{bmatrix} \mathbf{U}_1 \\ \overline{\mathbf{U}}_1 \\ \mathbf{U}_2 \\ \overline{\mathbf{U}}_2 \\ \mathbf{U}_3 \\ \overline{\mathbf{U}}_3 \end{bmatrix} = \begin{bmatrix} \mathbf{b}_1 & \mathbf{b}_2 & \mathbf{b}_3 & \dots & \mathbf{b}_{n-1} & \mathbf{b}_n \\ \overline{\mathbf{b}}_1 & \overline{\mathbf{b}}_2 & \overline{\mathbf{b}}_3 & \dots & \overline{\mathbf{b}}_{n-1} & \overline{\mathbf{b}}_n \\ \mathbf{b}_{i_1} & \mathbf{b}_{i_2} & \mathbf{b}_{i_3} & \dots & \mathbf{b}_{i_{n-1}} & \mathbf{b}_{i_n} \\ \overline{\mathbf{b}}_{i_1} & \overline{\mathbf{b}}_{i_2} & \overline{\mathbf{b}}_{i_3} & \dots & \overline{\mathbf{b}}_{i_{n-1}} & \overline{\mathbf{b}}_{i_n} \\ \mathbf{b}_{j_1} & \mathbf{b}_{j_2} & \mathbf{b}_{j_3} & \dots & \mathbf{b}_{j_{n-1}} & \mathbf{b}_{j_n} \\ \overline{\mathbf{b}}_{j_1} & \overline{\mathbf{b}}_{j_2} & \overline{\mathbf{b}}_{j_3} & \dots & \overline{\mathbf{b}}_{j_{n-1}} & \overline{\mathbf{b}}_{j_n} \end{bmatrix}, \tag{8}$$

where $\mathbf{b}_i \in \{0,1\}^{\lceil \log n \rceil}$ is the binary representation of $i-1$, the notation $\overline{\mathbf{b}}$ means the bitwise complement or the flipped sequence. Besides $(i_1, \dots, i_n)$, $(j_1, \dots, j_n)$ are independent random vectors, drawn uniformly at random from the set $[n]^n$, they represent the permutation vectors that permute the columns of the matrix as follows: The first position of vector i, let's say i_1 indicate that the $i_1 - th$ column will be the first column of U_2, the same holds for the other positions of i. Analogously the same procedure is applied to vector j, the blocks formed by $[\mathbf{U}_2, \overline{\mathbf{U}}_2]$ $[\mathbf{U}_3, \overline{\mathbf{U}}_3]$ are often named as permutation blocks. In order to clarify, we now give an example:

Remark 2. The name of adjacency matrix above does not match with the classical definition of adjacency matrix. The matrix is useful to represent bipartite matrix because we previously know that there are two kind of nodes that does not have any common edge. The reader should note the familiarity with the Tanner graph presented in the coding theory section, variable nodes and check nodes were replaced by bundle of tests and n total items.

Remark 3. The mathematical expression $diag(x)$ for a vector x means a diagonal matrix with diagonal equal to x

Example 4. Consider a group testing problem with $n = 3$ items. Suppose we construct a bipartite graph with adjacency matrix and permutation vectors i and j given below:

$$\mathbf{T} = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad i = (3,2,1) \quad j = (1,3,2)$$

We build the three blocks $\mathbf{U}_1$, $\mathbf{U}_2$ and $\mathbf{U}_3$ as follows:

$$\mathbf{U}_1 = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad \mathbf{U}_2 = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix} \quad \mathbf{U}_3 = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

Now we build the blocks $\overline{\mathbf{U}_1}$, $\overline{\mathbf{U}_2}$ and $\overline{\mathbf{U}_3}$ taking the flipped bit,i.e, the bit 0 flips to 1 and the bit 1 flips to 0.

$$\overline{\mathbf{U}_1} = \begin{bmatrix} 1 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix} \quad \overline{\mathbf{U}_2} = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix} \quad \overline{\mathbf{U}_3} = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix}$$

We now build the signature matrix $\mathbf{U}$ based on 8 :

$$\mathbf{U} = \left[\begin{array}{ccc} 0 & 1 & 0 \\ 0 & 0 & 1 \\ \hline 1 & 0 & 1 \\ 1 & 1 & 0 \\ \hline 0 & 1 & 0 \\ 1 & 0 & 0 \\ \hline 1 & 0 & 1 \\ 0 & 1 & 1 \\ \hline 0 & 0 & 1 \\ 0 & 1 & 0 \\ \hline 1 & 1 & 0 \\ 1 & 0 & 1 \end{array}\right]$$

We turn our attention now to construct the sensing matrix $\mathbf{A}$:

$$\mathrm{diag}(\mathbf{t}_1) = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad \mathrm{diag}(\mathbf{t}_2) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad \mathrm{diag}(\mathbf{t}_3) = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

The corresponding group testing matrix is given by three blocks of $\mathbf{A}_i = \mathbf{U}\ \operatorname{diag}(\mathbf{t}_i)$, $i \in \{1, 2, 3\}$:

$$\mathbf{A} = [\mathbf{A}_1 | \mathbf{A}_2 | \mathbf{A}_3]^T = \left[\begin{array}{ccc}
0 & 1 & 0 \\
0 & 0 & 0 \\
0 & 0 & 0 \\
0 & 1 & 0 \\
0 & 1 & 0 \\
0 & 0 & 0 \\
0 & 1 & 0 \\
0 & 0 & 0 \\
0 & 0 & 0 \\
0 & 1 & 0 \\
0 & 1 & 0 \\
0 & 0 & 0 \\
\hline
0 & 1 & 0 \\
0 & 0 & 0 \\
1 & 0 & 0 \\
1 & 1 & 0 \\
0 & 1 & 0 \\
1 & 0 & 0 \\
1 & 0 & 0 \\
0 & 1 & 0 \\
0 & 0 & 0 \\
0 & 1 & 0 \\
1 & 1 & 0 \\
1 & 0 & 0 \\
\hline
0 & 0 & 0 \\
0 & 0 & 1 \\
1 & 0 & 1 \\
1 & 1 & 0 \\
0 & 0 & 1 \\
0 & 0 & 0 \\
0 & 0 & 0 \\
0 & 0 & 0 \\
0 & 0 & 1 \\
0 & 0 & 0 \\
0 & 0 & 0 \\
0 & 0 & 1
\end{array}\right]$$

In order simplify the notation we define the observation vector corresponding to the right node i denoted by z_i:

Definition 13. The right node measurement corresponding to a right node i is defined to be:

$$\mathbf{z}_i = \mathbf{U} \odot \operatorname{diag}(\mathbf{t}_i)\mathbf{x} = \bigvee_{(\mathbf{t}_i)_j = 1, x_j = 1} u_j,\ 1 \leq i \leq M \tag{9}$$

The observation vector can be rewritten as:

$$\mathbf{y} = \begin{bmatrix} \mathbf{z}_1 \\ \vdots \\ \mathbf{z}_M \end{bmatrix}. \tag{10}$$

Remark 4. The observed vector in the above equation is exactly the same as the previous one. There are two ways to represent it, however we will show that is much more practical to represent as a matrix of right node measurements.

We write $\mathbf{u}_j$ to refer to the column j of the signature matrix $\mathbf{U}$. The signature matrix can be written with $[u_1\ u_2\ ...\ u_n]$ in terms of column vectors

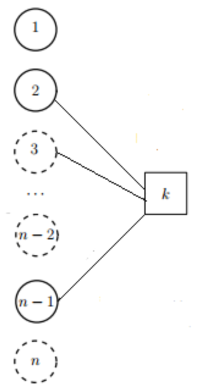

Figure 10: **Right Node Measurement Illustration**

Now consider an illustration of right node measurement, in which the dashed lines represents the non defective items, the associate right node measurement z_k is given by:

$$z_k = (u_2 \vee u_{n-1})$$

We can observe that the right node measurement is related with the number of defective items tested. Our decoding scheme will be based on this fact. We are going to present in the next section, the relation between the number of defective items being tested and how we can recover the indices of this items.

4 SAFFRON Algorithm

a Detecting and Resolving Singletons

We call a singleton as a right node connected with only one defective item. In this sense consider a signature matrix $\mathbf{V}$ given by:

$$\mathbf{V} = \begin{bmatrix} \mathbf{U}_1 \\ \overline{\mathbf{U}}_1 \end{bmatrix} = \begin{bmatrix} \mathbf{b}_1 & \mathbf{b}_2 & \mathbf{b}_3 & \dots & \mathbf{b}_{n-1} & \mathbf{b}_n \\ \overline{\mathbf{b}}_1 & \overline{\mathbf{b}}_2 & \overline{\mathbf{b}}_3 & \dots & \overline{\mathbf{b}}_{n-1} & \overline{\mathbf{b}}_n \end{bmatrix}, \tag{11}$$

Observe in this matrix $\mathbf{V}$ each column has the same Hamming weight $L = log_2 n$ corresponding to the vector length, the reason for it is due to each column is the sum of a binary vector and its complement, the column weight is the sum of two vectors, if the vector has 1 in a certain position the complement(flipped vector) has 0 at the same corresponding position and vice-versa, therefore we are summing 1 L times. According to the equation 9 if the node is a singleton it's measurement is a single column only, that implies a the hamming weight equal to L.

Furthermore observe that if the node is not a singleton, there are at least two vectors on the "OR"operation 9. The key detail relies on the fact that if there are more than one defective item being measured, the hamming weight of the corresponding right node measurement will never be equal to L. We now write the formal theorem:

Theorem 2. *Consider a group testing problem with n defective items. A right node is singleton if and only if the hamming weight of the corresponding right node measurement is equal to 3$log_2 n$*

Proof. Suppose that the node is a singleton, as we explained for a signature matrix V the hamming weight of the column is the same for all columns and it is equal to L. Since the a signature matrix U is composed by three matrices with the same type as V, we conclude that the hamming column weight is three times L. Applying the equation 9 with only one defective item,i.e, for only one position j, $\mathbf{x}_j = \mathbf{t}_j = 1$, the resulting right node measurement is u_j and since it is a single column it has hamming weight equal to 3L.
Conversely suppose that the right node measurement $\mathbf{z}_i$ has a hamming weight equal to 3L .Suppose by contradiction that it is not a singleton, therefore there are at least two vectors such that:

$$\mathbf{z}_i = u_1 \vee u_2 \vee ...$$

Since u_1 and u_2 are different, at least one position is different. Suppose that they differ at the $i-th$ position, observe the operator OR at this position will be 1 and the same holds for the operation OR between the position corresponding to the flipped bit. Therefore the vector $u_1 \vee u_2$ will have more ones than both u_1 and u_2, it implies that the hamming weight will be slightly larger than $3L$, it contradicts our assumption. □

The first step of the detection scheme is to compute the hamming weight of the right node measurement and check if it is equal to $3L$. If it is equal to $3L$, read the first $log_2 n$ positions of the right node measurement to identify the corresponding defective item.

Remark 5. Now we can see why the signature matrix contains a matrix $\mathbf{U}_1$ and its complement $\overline{\mathbf{U}}_1$ it is only to have exactly the same hamming column weight. Moreover observe that the permutation blocks are not important to detect singletons

Example 5. Consider a group testing problem with $n = 4$ items where $K = 2$ defectives, suppose the vector $\mathbf{x} = (0, 1, 1, 0)$, $i = (4, 2, 3, 1)$ and $j = (1, 3, 2, 4)$. The adjacency matrix T and the signature matrix U given below:

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

$$\mathbf{U} = \left[\begin{array}{cccc} 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ \hline 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ \hline 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 \\ \hline 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ \hline 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ \hline 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \end{array}\right]$$

The corresponding right nodes measurements are:

$$\mathbf{y} = \begin{bmatrix} \mathbf{z}_1 \\ \mathbf{z}_2 \end{bmatrix} = \begin{bmatrix} \mathbf{u}_3 \\ \mathbf{u}_2 \vee \mathbf{u}_3 \end{bmatrix}$$

$$\mathbf{z}_1 = (1,0|0,1|1,0|0,1|0,1|1,0) \quad \mathbf{z}_2 = (1,1|1,1|1,1|1,1|1,1|1,1)$$

We now take the hamming weight:

$$W(\mathbf{z}_1) = 6 \;\; W(\mathbf{z}_1) = 12$$

Observe that $3L = 3(log_2 4) = 6$, then only $\mathbf{z}_1$ is a singleton. The first two positions of the vector $\mathbf{z}_1$ are $(1,0)$ and its correspond to the third column of the signature matrix, therefore we conclude that the item 3 is defective.

Remark 6. If the number of items n is not a power of 2, we have to complete the defective vector $\mathbf{x}$ with zeros.

b Detecting and Resolving Doubletons

A node is called doubleton if it is connected with exactly two defective items. Consider the signature matrix $\mathbf{V}$ defined 11, the strategy to detect doubletons is not deterministic, in other words, instead of a theorem that guarantee us a perfect detection, SAFFRON offers a scheme that detect doubletons with high probability. In this context, the strategy is described below as an algorithm:

Algorithm 1 Detecting and Resolving Doubletons

```
Step1: Detect all singletons and the corresponding defective items using the strategy described in the previous section
Step2: Identify all possible doubletons, all nodes connected with a defective item but is not a singleton
for Each possible doubleton do
    Step3: Divide the corresponding right node measurement in three blocks with same length $\mathbf{z} = (\mathbf{z}^1; \mathbf{z}^2; \mathbf{z}^3)$, the same for the column vector corresponding to the defective item $\mathbf{u}_k = (\mathbf{u}_k^1; \mathbf{u}_k^2; \mathbf{u}_k^3)$
    Step4: For an unknown vector $\mathbf{u}_l = (\mathbf{u}_l^1; \mathbf{u}_l^2; \mathbf{u}_l^3)$ solve the following equations : $\mathbf{z}^i = \mathbf{u}_k^i \vee \mathbf{u}_l^i \; i \in \{1,2,3\}$
    Step5: Check if the vector $\mathbf{u}_l$ can be a column of the signature matrix $\mathbf{U}$
    Step6: In affirmative case, declare the node as a doubleton and find the corresponding defective item. Otherwise declare as unsolvable node
    Step7: Update the set of possible doubletons.
end for
```

We should consider two mathematical details. The first one is about the step 4, we have to ensure that all logical equations of this type will be uniquely solvable. Next, what is the probability to declare a wrong doubleton and identify a non defective item as defective. We now introduce two theorems related with these problems

Theorem 3. *Let $\mathbf{z}$ be a right node measurement and let u_k be a column vector of the signature matrix $\mathbf{U}$. Then the equation*

$$\mathbf{z} = u_k \vee u_l$$

has unique solution if the vector is given by $\mathbf{u}_l = (\mathbf{b}_1; \overline{\mathbf{b}}_1; \mathbf{b}_2; \overline{\mathbf{b}}_2; \mathbf{b}_3; \overline{\mathbf{b}}_3)$

Proof. Observe that for each vector position we have two possibilities:

$$\mathbf{z}(i) = 1 \vee \mathbf{u}_l^1(i) \; or \; \mathbf{z}(i) = 0 \vee \mathbf{u}_l^1(i)$$

We can only figure out the unknown bit in the second case, because in the first case the result of the logical operation is always the bit 1 independent of the unknown bit. The key idea to always have solution is to observe that if we have in the i-th position the first case, we will have the second case when we take the complement so we solve the equation for the complement. We conclude that for every position we can find the unknown bit. □

Theorem 4. *SAFFRON resolves all doubletons and when a node is connected with more than two defective items, SAFFRON declares a wrong doubleton with probability no greater than $\frac{1}{n^2}$ in which n is the number of total items.*

Proof. From the previous theorem it is clear that SAFFRON detect and solve all doubletons. Now consider a right node connected with more than two defective items, SAFFRON declares a wrong defective item if item l_1 associated with the vector $\mathbf{u}_l^1$ is not connected with the node and two logical equations with the permutation indices hold. In this sense, observe that when the item l_1 is not connected, i_{l1} and j_{l2} are independent from the corresponding l_2 and l_3, the probability to have $l_2 = il_1$ and $l_3 = jl_1$ is $\frac{1}{n^2}$ □

Corollary 1. *SAFFRON recovers all defective items with probability greater or equal to $1 - \mathcal{O}(\frac{K}{n^2})$.*

Proof. We apply the union bound probability, observe that according to the previous theorem SAFFRON declares a wrong defective items with probability no greater than $\frac{1}{n^2}$, since it is well known the total number of defective items K, we use the union bound to bound the probability of error. Observe that the algorithm iterates $\mathcal{O}(K)$ times, therefore we have a perfect detection with probability $1 - \mathcal{O}(\frac{K}{n^2})$ □

Remark 7. Now we can also see why the signature matrix contains two permutations blocks, it is helpful to declare a wrong doubleton with low probability, moreover, if we increase the number of copies we decrease the probability of error. Of course, increasing the number of copies increase the computational complexity as we will note.

Example 6. Consider the previous example 5. We declared the item 3 as defective, we now note that the second right node is probable to be a doubleton. We divide $\mathbf{z}_2$ and $\mathbf{u}_3$ in three blocks :

$$\mathbf{z}_2^1 = (1,1|1,1) \;\; \mathbf{z}_2^2 = (1,1|1,1) \;\; \mathbf{z}_2^3 = (1,1|1,1)$$

$$\mathbf{u}_3^1 = (1,0|0,1) \;\; \mathbf{u}_3^2 = (1,0|0,1) \;\; \mathbf{u}_3^3 = (0,1|1,0)$$

We write $\mathbf{z}_2^1 = \mathbf{u}_l^1 \vee \mathbf{u}_3^1$ and we obtain $(1,1|1,1) = (1,0|0,1) \vee (b_1, b_2|\bar{b}_1, \bar{b}_2)$ we recover the vector $(b_1, 1|1, \bar{b}_2)$ and observe since $\bar{b}_1 = 1$ then $b_1 = 0$ and $b_2 = 1$ then $\bar{b}_2 = 0$, to conclude the first part we note that the first block is given by $u_l^1 = (0,1|1,0)$.

We apply the same procedure for the others vectors and obtain the vector $u_l = (0,1|1,0|0,1|1,0|1,0|0,1)$. Now we check if this vector can be a column of the signature matrix, u_l is the second column of the signature matrix, we conclude that the second right node is a doubleton and the item 2 is defective.

c Decoding

The decoding scheme is based on two steps. The first one is to detect defective items associated with singletons. When it is done, apply the algorithm to find doubletons until recover the total set of defective items or stop when the set of updated doubletons is the same as the non-updated, it implies there is a defective item that cannot be identified. We now give a full example of the SAFFRON algorithm we are going to omit details of the singleton and doubleton procedures because they were explained before.

Remark 8. The following example is a toy example, it is not useful for practical applications when you have a small number of total items. For group testing algorithms we are always interested in the asymptotic case, when n goes to infinity to be precisely

Remark 9. When we are simulating a group testing algorithm we suppose to have the answer, but we only use the result of tests to recover and check if our recovered vector matches with the vector.It happens due to we do not have physical devices to collect real data, it is a very common approach to simulate inverse problems, problems in which you want to estimate or decode an input collecting data from the output of a device.

Example 7. Consider a group testing situation with $n = 8$ total items and $K = 3$ defective items. Suppose the vector of defective items is $x = (1, 0, 1, 0, 0, 0, 0, 1)$. Suppose we have the following adjacency matrix with $M = 4$ right nodes:

$$\mathbf{T} = \begin{bmatrix} 0 & 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 1 & 1 & 0 & 1 \end{bmatrix} \in \{0, 1\}^{M \times n}$$

Now assume we have the following random sequences for the permutation vectors:

$$(i_1, i_2, ..., i_8) = (5, 2, 4, 8, 7, 1, 3, 6) \;\; (j_1, j_2, ..., j_8) = (3, 1, 5, 6, 3, 8, 2, 7)$$

The signature matrix $\mathbf{U}$ is given by:

$$\mathbf{U} = \left[\begin{array}{cccccccc} 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ \hline 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ \hline 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ \hline 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ \hline 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ \hline 1 & 1 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{array}\right]$$

Using the equation 9 we have the following right node measurements:

$$\mathbf{y} = \begin{bmatrix} \mathbf{z}_1 \\ \mathbf{z}_2 \\ \mathbf{z}_3 \\ \mathbf{z}_4 \end{bmatrix} = \begin{bmatrix} \mathbf{u}_3 \\ \mathbf{u}_1 \vee \mathbf{u}_3 \vee \mathbf{u}_8 \\ \mathbf{u}_1 \vee \mathbf{u}_8 \\ \mathbf{u}_3 \vee \mathbf{u}_8 \end{bmatrix} \tag{12}$$

To begin with, we shall observe the right node measurement vectors:

$$\mathbf{z}_1 = (0,1,0|1,0,1|0,1,1|1,0,0|1,0,0|0,1,1)^T$$

$$\mathbf{z}_2 = (1,1,1|1,1,1|1,1,1|1,1,1|1,1,0|1,1,1)^T$$

$$\mathbf{z}_3 = (1,1,1|1,1,1|1,0,1|0,1,1|1,1,0|1,0,1)^T$$

$$\mathbf{z}_4 = (1,1,1|1,0,1|1,1,1|1,1,0|1,1,0|0,1,1)^T$$

$$W(\mathbf{z}_1) = 9 \;\; W(\mathbf{z}_2) = 17 \;\; W(\mathbf{z}_3) = 14 \;\; W(\mathbf{z}_4) = 14$$

Observe that only $\mathbf{z}_1$ has hamming weight equal to $3log_2n$. According to the theorem 3, the right node 1 is a singleton. Then we read the first three bits of $\mathbf{z}_1$ $(0,1,0)$ and observe that it corresponds to the third column. We conclude that the item 3 is defective and finish the first step because we do not have singletons anymore.

Now we start the second step. The possible doubletons are the right nodes connected with the defective item 3,i.e, the right node 2 and 4, observe that $\mathbf{T}_{3,3} = 0$, the right node 3 is not connected to the defective item 3. We guess that the right node 2 is a doubleton, we write the equations according to the step 4 and solve them:

$$\mathbf{u}_l^1 = (101|010) \;\; \mathbf{u}_l^2 = (100|011) \;\; \mathbf{u}_l^3 = (010|101) \;\; \mathbf{u}_l = (101|010|100|011|010|101)$$

Observe if the vector $\mathbf{u}_l$ is a column vector, searching for $\mathbf{u}_l^1$ in the signature matrix $\mathbf{U}$ we find that the corresponding column is 6. Doing the same procedure with $\mathbf{u}_l^2$, we find that the corresponding column is 1. Observe that the conclusion does not match, $\mathbf{u}_l$ cannot be a column vector, therefore the right node 2 is not resolvable.

As we mentioned we perform the same thing with the next possible doubleton. We write the vectors again, at this time, for the right node 4:

$$\mathbf{u}_l^1 = (111|000) \;\; \mathbf{u}_l^2 = (101|010) \;\; \mathbf{u}_l^3 = (110|001) \;\; \mathbf{u}_l = (111|000|101|010|110|001)$$

Searching for $\mathbf{u}_l^1$, we find the column 8, searching for $\mathbf{u}_l^2$ we also find the column 8 and searching for $\mathbf{u}_l^3$ we find again the column 8! We declare the right node 4 as a doubleton and identify 8 as a defective item.

Updating the set of possible doubletons, we observe that we have a new possible doubleton, right node 3 is connected with the defective item 8 and is not a singleton. We repeat the same steps as before and write the following vectors for the right node 3:

$$\mathbf{u}_l^1 = (000|111) \;\; \mathbf{u}_l^2 = (100|011) \;\; \mathbf{u}_l^3 = (010|101) \;\; \mathbf{u}_l = (000|111|100|011|010|101)$$

We also search for the vectors and conclude that $\mathbf{u}_l$ is exactly the first column of the signature matrix $\mathbf{U}$, so we declare right node 3 as a doubleton and recover the defective item 3.

Now observe that we do not have any iteration anymore. We recovered three defective items and the total number of defective items $K = 3$, the recovered vector is $(1, 0, 1, 0, 0, 0, 0, 1)$ and is exactly the same the target vector x. We conclude that SAFFRON recovered all defective items.

Remark 10. For practical implementations, we can check if the recovered vector $\mathbf{u}_l$ is a vector of the signature matrix $\mathbf{U}$ as follows: Read off the first three bits of each vector, u_l^1,u_l^2 and u_l^3 and check if it match with the permutation vector. For a concrete example, consider the vectors obtained in the previous example related to the right node 2. Read the first three bits of the first vector represents 6, the second represents 5 and third 3. Finally observe that $i_6 = 1 \neq 5$ and $j_6 = 8 \neq 3$, it does not match so it cannot be a doubleton

5 Analyzes of SAFFRON Scheme

At this moment we described the decoding scheme and prove some guarantees. Nevertheless, still missing the results about computational complexity and the design of the bipartite graph.

a The Main Theorem

The main theoretical result of SAFFRON is the following theorem:

Theorem 5. *Given an error floor ϵ. With $m = 6C(\epsilon)K\log n$ tests, SAFFRON algorithm recovers at least $(1-\epsilon)K$ defective items with probability $1 - \mathcal{O}(\frac{K}{n^2})$, where ϵ is an arbitrarily-close-to-zero constant and $C(\epsilon)$ is a constant that only depends on ϵ.*

Proof. First observe that each right node is connected with 6$log_2 n$ tests because each right node measures according 9, so we need to use all rows of the signature matrix $\mathbf{U}$, we have $log_2 n$ rows to write n in binary, we always assume n to be a power of 2. Since we duplicate the rows by flipping the bits we now have 2$log_2 n$ and we build three blocks so we have at the end 6$log_2 n$.

We need now to prove that the number of required right nodes to achieve our goal is given by a constant that only depends on ϵ times K. As we mentioned we construct a d-regular right node graph draw uniformly at random. We focus on the pruned subgraph consisting in K defective left nodes and the same right nodes then the average right degree $\lambda = \frac{Kd}{M}$, further as K increases we can approximate the degree distribution of right nodes with a Poisson distribution. We recall that the edge distribution is given by $\sum_{i=1}^{\infty} \rho_i x^{i-1}$ where ρ_i is the probability that a randomly selected edge in the graph is connected to a right node of degree i. We have the following identities:

$$\rho_i = \frac{iM}{Kd} Pr(degree\ of\ a\ random\ right\ node = i) = \frac{iM}{Kd} e^{-\lambda} \frac{\lambda^i}{i!} = e^{-\lambda} \frac{\lambda^{i-1}}{(i-1)!}$$

The fraction of unidentified items can be analyzed by density evolution. In the section about coding theory we explained that density evolution is a tool to analyze message passing algorithms. Let p_j be the probability that a defective item not be identified at the iteration j, the main idea to use density evolution is to relate p_j to p_{j+1} as follows:

$$p_{j+1} = Pr(not\ resolvable\ from\ one\ children\ right\ node = i)^{d-1} = (1 - (\rho_1 + \rho_2(1-p_j)))^{d-1} \tag{13}$$

Where $\rho_1 = e^{-\lambda}$ and $\rho_2 = \lambda e^{\lambda}$. To see why it is true, at iteration $j+1$ the left node v returns a not recovered message to right node c if none of its other neighbor right nodes $\{c_i\}_{i=1}^{d-1}$ has been identified as either a singleton or a resolvable doubleton at iteration j. The probability that one of this right nodes be solved by a singleton or doubleton is $\rho_1 + \rho_2(1-p_j)$ because the node can be a singleton, so it connected to a right node of degree one or be a doubleton, i.e, connected with a right with degree two and not resolved at previous iterations. Then given a tree neighborhood of v, all messages are independent, then the equation 13 holds.
To characterize the fraction of defective items that will not be recovered we take the limit, we define the error floor ϵ to be the limit of p_j as $j \to \infty$. To simplify the notation we write $\gamma(p) = (1 - (\rho_1 + \rho_2(1-p)))^{d-1}$. The strategy is to choose λ in order to make γ a contraction, then we will have a unique fixed point according to the theorem 1.
We now design an optimal pair (d, M) that minimizes the number of right nodes M given a target reliability ϵ. We approximately compute the fixed point to be:

$$\epsilon = (1 - e^{\lambda} - \lambda e^{\lambda})^{d-1} \tag{14}$$

Then we solve the following optimization problem:

$$\begin{aligned} \min_{\substack{\lambda>0 \\ d\in\mathbb{N}}} \quad & M = \frac{Kd}{\lambda} \\ \text{subject to} \quad & (d-1)\log\left(1-e^{\lambda}-\lambda e^{-\lambda}\right) \leq \log(\epsilon) \\ & 1-e^{-\lambda}-\lambda e^{-\lambda} < \lambda. \end{aligned}$$

The first constraint is to guarantee the target reliability and the last constraint is to ensure that the equation is a contraction. The optimal points λ^* and d^* are obtained as function of the error floor ϵ. Moreover our constant $C(\epsilon)$ is defined as $C(\epsilon) = \frac{M}{K} = \frac{d^*}{\lambda^*}$.

To finish the proof we need to prove that with high probability, a constant deep neighborhood of a random left node is a tree and combine this result with the contraction strategy to show that the actual fraction of unidentified defective item is highly concentrated around its average. The proofs are very technical and can be found in [23].

$\square$

Here an example of fixed point scheme. Observe how the γ function intersects the blue line only once.

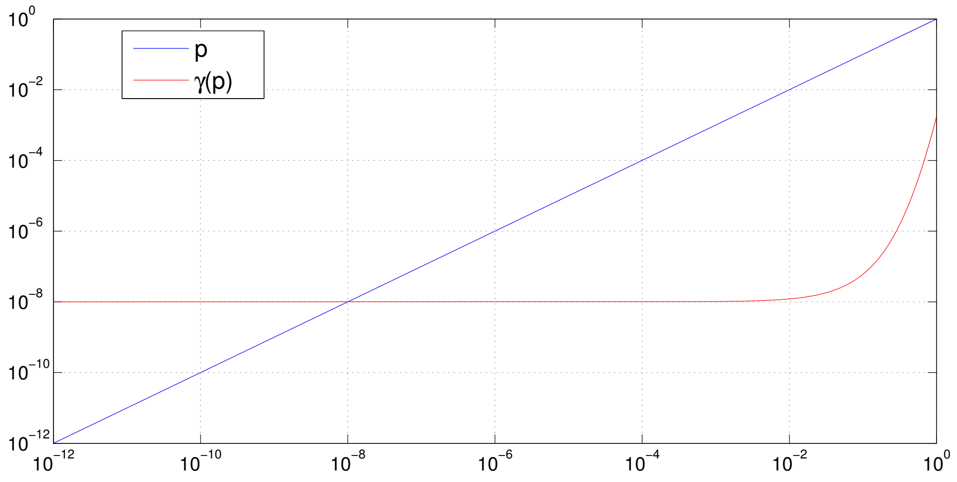

Figure 11: **Fixed Point Illustration**
We considered here $\lambda = 1.02$ and $d^* = 15$. Observe that the target reliability found is exactly 10^{-8}. Therefore our fixed point is the unique point when the red curve intersects the blue line

The next step is to find a solution of the optimization problem in the proof.

b Optimization Problem

Motivated by the optimization problem we write a brief introduction to techniques related to linear programming. Although the optimization problem can be solved numerically using a heuristic method, we offer an alternative method to solve it faster.

To begin with the strategy to solve the optimization problem, we prove the following inequality:

Lemma 1. $\forall x > 0$. *We have* $1 - x \leq e^{-x}$

Proof. Observe that the function $f(x) = e^{-x} + x - 1$ has the derivative equal to $f'(x) = -e^{-x} + 1 \geq 0$ so the function is monotone increasing. □

We use this inequality as a tool to linearize our constraint.

$$(1 - e^{-\lambda} - \lambda e^{-\lambda}) \leq (-\lambda + \lambda^2) + (-1 + \lambda) + 1 \leq \lambda^2$$

Using the same strategy in the lemma, observe that for $\lambda > 0$ we have the last constraint always satisfied. Using the fact that minimize λ is the same as minimize $\log(\lambda)$, we can write the equivalent model:

$$\begin{aligned} \min_{\substack{\lambda' > 0 \\ d \in \mathbb{N}}} \quad & M = \frac{d}{\lambda'} \\ \text{subject to} \quad & (d-1)\lambda' \leq \epsilon' \end{aligned}$$

Where $\lambda' = log(\lambda)$ and $\epsilon' = \frac{log \epsilon}{2}$. Now we have a fractional linear programming, since d is a discrete parameter we can write it as a linear constraint. This model can be solved using a standard linear programming algorithm with change of variables, for more details read [24].

We now present a table of constants $C(\epsilon)$ and optimal d^* according to a given error floor ϵ. Solving the optimization problem we have:

ϵ	10^{-3}	10^{-4}	10^{-5}	10^{-6}	10^{-7}	10^{-8}	10^{-9}	10^{-10}
$C(\epsilon)$	6.13	7.88	9.63	11.36	13.10	14.84	16.57	18.30
d^*	7	9	10	12	14	15	17	19

Table 2: **Constant $C(\epsilon)$ vs. error floor ϵ**

Remark 11. The computational complexity of the optimization algorithm is not part of the decoding complexity, since we can solve once for all and save the values in a table just as we did above

6 Simulation of SAFFRON

According to the main theorem SAFFRON recover at least $(1-\epsilon)K$ defective items with probability $1-\mathcal{O}(\frac{K}{n^2})$. We wrote a computer program in matlab code to simulate the performance. Precisely we compute the fraction of unidentified items using the hamming distance defined in the coding theory section divided by the number of total items. In digital communication language we simulate the bit error rate (BER). We present our results, first a table related to 2, for the others values we obtained a perfect reconstruction. We performed a monte carlo simulation

$K\epsilon$	10^{-2}	10^{-3}	10^{-4}	10^{-5}
BER	$2\ 10^{-3}$	$7.8\ 10^{-4}$	$9\ 10^{-5}$	$5\ 10^{-5}$

Table 3: **Fraction of Unidentified items x Error Target**
$n = 2^8$, $K = 10$, 100 iterations Monte Carlo Simulation

We also simulate a graph relating the average fraction of unidentified according to many values of $\frac{M}{K}$. The idea is to evaluate the performance of SAFFRON using a very sparse graph.

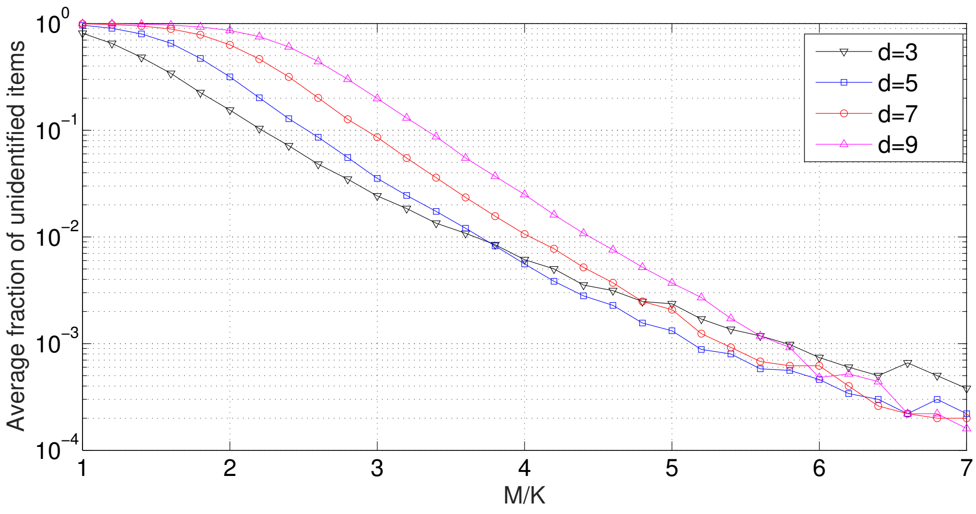

Figure 12: **The average fraction of unidentified defective items**
$n = 2^{16}$, $K = 100$, 100 iterations of Monte Carlo Simulation

We can conclude that behaves well even if we have a very sparse graph. Besides, we evaluate the algorithm complexity for a range of values of K and compare with the naive strategy, test every item. The algorithm complexity of the greedy strategy is, of course, exactly equal to n. In order to have a fair comparison, the greedy strategy recovers all items with no probability of error, so we considered a constant $C(\epsilon)$ corresponding to a very low error floor $\epsilon \leq 10^{-10}$:

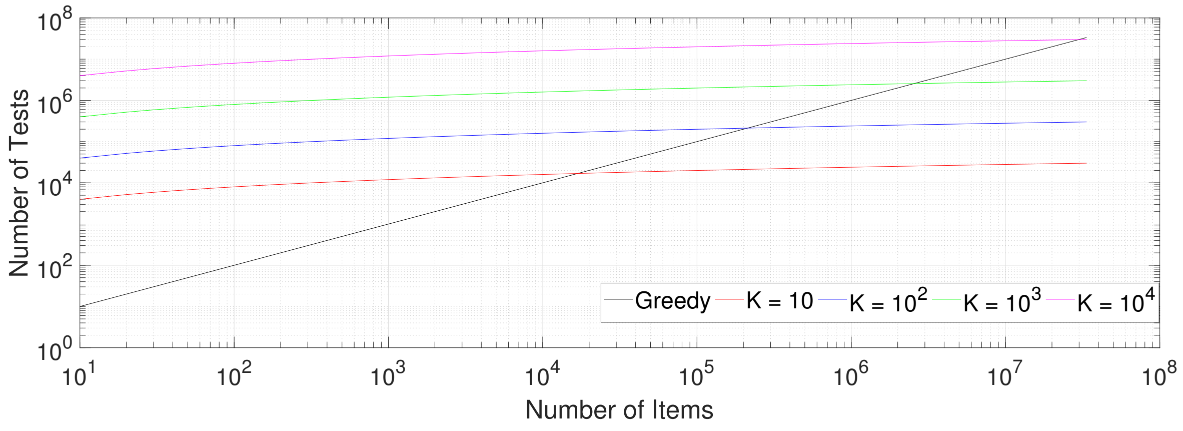

Figure 13: **Algorithm Complexity Evaluation**

Clearly SAFFRON is not adequate for practical applications for a certain range of n. Precisely when the number of items is small enough(depending on K) the greedy strategy has the same or a similar computational complexity and therefore it is not useful.

7 Generalizations

In this section we present some generalizations of the group testing problem, we want to present the readers future directions of research in group testing theory and related problems. In this sense we present general formulation of the problems and a sketch of the strategy to solve them.

a Noisy Group Testing

Noisy group testing problem is considered when we have devices with a low signal noise ratio, we include uncertainty in the measurements, so the observed vector $\mathbf{y}$ is now modified

$$\mathbf{y} = \mathbf{A} \odot \mathbf{x} \oplus \mathbf{w} \tag{15}$$

Where $\mathbf{w}$ is the noise vector that assumes the value 1 with probability q and assumes 0 with probability $1-q$, the symbol $\oplus$ denote the logical operation XOR. The approach to solve this problem is to design a signature matrix U' consisted of encoded columns. The idea is to simulate a message through a noisy memory less channel and apply a modern LDPC decoder called Spatial Coupled low density parity check decoder.

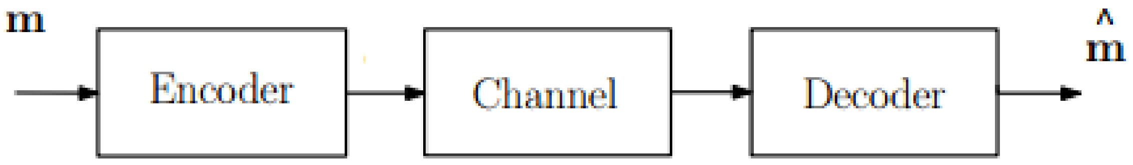

Figure 14: **Channel transmission scheme**

A theorem in [25] ensures that exists general transmission scheme with spatial coupled LDPC having the following properties:

- An encoding function $f : \{0,1\}^N \to \{0,1\}^{\frac{N}{R}}$
- A decoding function $g : \{0,1\}^{\frac{N}{R}} \to \{0,1\}^N$ with decoding complexity $\mathcal{O}(N)$
- if R satisfies the following inequality:

$$R < q\ log_2 q + (1-q)\ log_2(1-q) - \delta$$

For an arbitrary small $\delta > 0$, then the probability to have a $g(\mathbf{x}+\mathbf{w}) \neq x$ less than $2^{-\zeta n}$ for some $\zeta > 0$.

The signature matrix $U' \in \{0,1\}^{\frac{6log2_n}{R} \times n}$ is given below:

$$\mathbf{U}' = \begin{bmatrix} \frac{f(\mathbf{b}_1)}{\overline{f(\mathbf{b}_1)}} & \frac{f(\mathbf{b}_2)}{\overline{f(\mathbf{b}_2)}} & \frac{f(\mathbf{b}_3)}{\overline{f(\mathbf{b}_3)}} & \dots & \frac{f(\mathbf{b}_{n-1})}{\overline{f(\mathbf{b}_{n-1})}} & \frac{f(\mathbf{b}_n)}{\overline{f(\mathbf{b}_n)}} \\ \frac{\mathbf{b}_{i_1}}{\overline{f(\mathbf{b}_{i_1})}} & \frac{\mathbf{b}_{i_2}}{\overline{f(\mathbf{b}_{i_2})}} & \frac{\mathbf{b}_{i_3}}{\overline{f(\mathbf{b}_{i_3})}} & \dots & \frac{\mathbf{b}_{i_{n-1}}}{\overline{f(\mathbf{b}_{i_{n-1}})}} & \frac{\mathbf{b}_{i_n}}{\overline{f(\mathbf{b}_n)}} \\ \frac{\mathbf{b}_{j_1}}{\overline{f(\mathbf{b}_{j_1})}} & \frac{\mathbf{b}_{j_2}}{\overline{f(\mathbf{b}_{j_2})}} & \frac{\mathbf{b}_{j_3}}{\overline{f(\mathbf{b}_{j_3})}} & \dots & \frac{\mathbf{b}_{j_{n-1}}}{\overline{f(\mathbf{b}_{j_{n-1}})}} & \frac{\mathbf{b}_{j_n}}{\overline{f(\mathbf{b}_{j_n})}} \end{bmatrix}, \tag{16}$$

We now sketch the robustified SAFFRON scheme that is very similar to the SAFFRON decoding scheme. We recall that the first step is to detect singleton, to detect singleton we divide the measurement vector into six blocks, divide first in three blocks as explained in doubleton section and after that divide by two each block creating a sub-block and the complement of this sub-block. We take the first, third and fifth segment of this division, $g(z_k^1)$, $g(z_l^3)$ and $g(z_k^5)$. We also write $l_1 - 1$ to be the decimal representation of $g(z_k^1)$, $l_2 - 1$ decimal representation for $g(z_l^3)$ and finally $l_3 - 1$ the same for $g(z_k^5)$. The detection rule is to check:

$$i_{l_1} = l_2 \quad j_{l_1} = l_3$$

Where i and j are the permutation vectors. The doubleton scheme is the same as SAFFRON only replacing the signature matrix. To end up with the noisy group testing section we present without proof the main results, we refer the readers ? for more details and proofs.

Theorem 6. *Robustified SAFFRON misses a singleton with probability no greater than* $\frac{3}{n^\zeta}$ *and declares a wrong defective item with probability no greater than* $\frac{1}{n^{2+\zeta}}$

Remark 12. Note that in the noisy group testing we have a probability of error in detecting singletons. In the SAFFRON algorithm the detection of singleton was without any errors.

This theorem is analogue version of the theorem of 4. Now we present the analogue version of the main theorem 5.

Theorem 7. *With* $m = 6\beta(q)C(\epsilon)Klog_2n$ *tests, Robustified SAFFRON can recover at least* $(1-\epsilon)K$ *defective items with probability* $1 - \mathcal{O}(\frac{K}{n^{2+\zeta}})$, *where* ϵ *is an arbitrary close to zero constant,* $C(\epsilon)$ *is a constant that depends only on the target reliability and* $\beta(q) = \frac{1}{R}$

b Multi Level Group Testing

Multi level tests can be viewed as a generalization of the test in group testing problem. The test does not return only zero and one anymore, it has intermediate levels, for example, the multi access channel problem presented in the introductory section considers a test that returns 0, 1 or 2. In this sense the authors [12] considered the algorithm for an arbitrary number of logical levels L, formally the test can be written as follows, consider the same notations as in the SAFFRON :

Definition 14. The multilevel test is defined :

$$\langle \mathbf{a}_i, \mathbf{x} \rangle_L = \min\{\sum_{j=1}^{n} a_{ij}x_j, L\}. \tag{17}$$

Remark 13. In the Multi Access Channel problem, the number of logical levels L is equal to 2

The SAFFRON version with the multi level test is called MULTI SAFFRON. The decoding scheme is exactly the same as SAFFRON with one modification, we do not decode until doubletons. We refer an l-ton to be a node connected with exactly l defective items. The decoding scheme will perform the same measurements as SAFFRON and stop decoding in the $2L$-ton, in which L is the number of logical levels, the following theorem explains

Theorem 8. *Let L be the number of logical levels. Let $n \in \mathbb{N}$, suppose a right node (Test) is a n-Ton. Then MULTI SAFFRON solves a n-Ton if and only $n \leq 2L$*

Proof. Let's define u_n as the vector we want to solve. In order to recover the vector u_n we need to ensure that each coordinate i will have an unique solution. Suppose we have a n-ton denoted by $u_1, .., u_n$ and a right node measurement z. We need to solve the following equation for every coordinate of u_n:

$$< u_1, ..., u_n >_L = z$$

We need to ensure that we will have $< u_1, ..., u_{(n-1)>_L} < L$, for each position, because if it is equal L the last bits u_n will not change the operation , so it is unsolvable. We apply the same strategy for the bit flipped equation $< \overline{u_1}, ..., \overline{u_{(n-1)}} >_L < L$. In order to guarantee that for every case one of the inequalities hold, we analyze the worst situation for each one, when we perform tests with only bit ones, it is the worst situation for the first inequality, we have the largest output L with the minimum number of bits required to achieve it. We repeat the same analysis for the second inequality, however the worst situation will be L bits zeros, because after flipped they achieve the largest output with the minimum number of bits required. Therefore if we cannot have L bits ones and L bits zeros, we bound $n - 1 < 2L$ that is equivalence to $n \leq L$

$\square$

For a more detailed version of the theorem and the corresponding proof see [12]. The main theorem version of MULTI SAFFRON algorithm is :

Theorem 9. *With $m = 6C(\epsilon, L)K \log n$ tests, our algorithm recovers at least $(1 - \epsilon)K$ defective items with probability $1 - \mathcal{O}(\frac{K}{n^2})$, where ϵ is an arbitrarily-close-to-zero constant and $C(\epsilon, L)$ is a constant that only depends on ϵ and L.*

To finish with this section we present the table of constants to compare with the SAFFRON :

ϵ	10^{-3}	10^{-4}	10^{-5}	10^{-6}	10^{-7}	10^{-8}	10^{-9}	10^{-10}
$C(\epsilon, L)$	2.30	3.01	3.84	4.05	4.60	5.50	5.85	6.73
d^*	4	4	4	6	7	6	12	7

Table 4: **Constant $C(\epsilon, L)$ vs. error floor ϵ for $L = 2$**

ϵ	10^{-3}	10^{-4}	10^{-5}	10^{-6}	10^{-7}	10^{-8}	10^{-9}	10^{-10}
$C(\epsilon, L)$	1.29	1.56	1.86	2.15	2.54	3.1	3.3	3.5
d^*	3	4	5	6	8	4	6	11

Table 5: **Constant $C(\epsilon, L)$ vs. error floor ϵ for $L = 3$**

As expected, with more logical levels we have lower constants.

8 Conclusions & future work

a Conclusions

In this work we presented the group testing and generalizations solved with tools from modern coding theory, to end up with we can conclude that :

- We carefully explained the proofs and provided teaching examples
- We connected group testing problem with a lot of applications in different branches of human knowledge, in special, applications in digital communication systems
- We explained how different areas such as modern coding theory, compressed sensing and signal processing were combined to create the best known algorithm for non adaptive probabilistic group testing
- We successfully reproduced SAFFRON algorithm
- We correctly generalized the algorithm for more general cases of group testing

b Future Works

Group Testing problem is a very deep problem with a lot of connections, it is natural that the problem has a lot of variations, generalizations and proposed algorithms from many different research communities. Despite a lot of directions to follow, we summarize some future research directions and the related open problems in the area.

- As we noted, SAFFRON algorithms achieves a considerable low computational complexity only when the number of total items is large enough, i.e, the algorithm works well in the asymptotic case. And what about the case when the number of total items is not so large but still large enough to be expensive to perform all tests, can we create a smart algorithm that works for all cases?
- How is the behaviour of MULTI SAFFRON algorithm in the presence of noise. Probably we can robustify the algorithm with a similar strategy used to create robustified SAFFRON, which guarantees we will have to recover the defective items
- In the literature, a new type of group testing named as threshold group testing were proposed by [26] [27], it was conjectured if the threshold group testing could be solved in the same order primal as the traditional group testing. The modern coding theory tools will be useful again to deal with this challenge?
- Sparse Recovery is a major challenge in the digital era. Can we extend the success of applying modern coding theory tools such as sparse graphs in other problems related with sparse recovery? What would be different if we apply sparse graph codes in compressed sensing, phase retrieval problems, logistic regression and high dimensional statistics or high dimensional signal estimation?

References

[1] K. Lee, R. Pedarsani, and K. Ramchandran, “Saffron: A fast, efficient, and robust framework for group testing based on sparse-graph codes,” *2016 IEEE International Symposium on Information Theory (ISIT)*, 2016.

[2] F. K. Hwang, *Combinatorial Group Testing and its applications*, 2000.

[3] H. Chen and F. Hwang, “A survey on nonadaptive group testing algorithms through the angle of decoding.” *Journal of Combinatorial Optimization*, vol. 15, 2008.

[4] A. Ganesan, S. Jaggi, and V. Saligrama, “Learning immune-defectives graph through group tests,” *IEEE ISIT*, 2015.

[5] M. Goodrich, M. Atallah, and R. Tamassia, “Indexing information for data forensics,” *Applied Cryptography and Network Security*, vol. 3531, 2005.

[6] M. T. Goodrich and D. S. Hirschberg, “Improved adaptive group testing algorithms with applications to multiple access channels and dead sensor diagnosis,” *Journal of Combinatorial Optimization*, vol. 15, 2008.

[7] A. Emad and O. Milenkovic, “Poisson group testing: A probabilistic model for nonadaptive streaming boolean compressed sensing,” *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2014.

[8] A. C. Gilbert, M. A. Iwen, and M. J. Strauss, “Group testing and sparse signal recovery,” *42nd Asilomar Conference on Signals, Systems and Computers*, 2008.

[9] B. K. Natarajan, “Sparse approximate solutions to linear systems,” *SIAM Journal of Computing*, vol. 24, 1996.

[10] J. Fürnkranz and T. Kliegr, “A brief overview of rule learning,” *Rule Technologies: Foundations, and its applications*, 2015.

[11] D. Malioutov and K. Varshney, “Exact rule learning via boolean compressed sensing ,” *Proceedings of The 30th ICML*, 2013.

[12] P.Abdalla, A.Reisizadeh, and R. Pedarsani, “Multilevel group testing via sparse graph codes,” *51st Asilomar Conference on Signals, Systems and Computers*, 2017.

[13] E. Newcommer and G. Lomow, *Understanding SOA with Web Services*, 2004.

[14] R. Diestel, *Graph Theory*, 2000.

[15] T. Richardson and R. Urbanke, *Modern Coding Theory*. Cambridge University Press, 2000.

[16] R. Gallager, “Low density parity check codes,” Ph.D. dissertation, MIT University, 1963.

[17] C. Healy, “Short-length low-density parity-check codes: Construction and decoding algorithms,” Ph.D. dissertation, University of York, 2014.

[18] M. J. Wainwright, “Stochastic processes on graphs with cycles: geometric and variational approaches,” Ph.D. dissertation, MIT University, 2002.

[19] A. J. Felström and K. S. Zigangirov, “Time-varying periodic convolutional codes with low-density parity-check matrix,” *IEEE Transactions of Information Theory*, vol. 45.

[20] A. G. D’yachkov and V. V. Rykov, “Bounds on the length of disjunctive codes,” *Problemy Peredachi Informatsii*, vol. 18, 1982.

[21] P. Indyk, H. Q. Ngo, and A. Rudra, “Efficiently decodable non-adaptive group testing,” *Proceedings of the Twenty-first Annual ACM-SIAM Symposium on Discrete Algorithms*, 2010.

[22] S. Cai, M. Jahangoshahi, M. Bakshi, and S. Jaggi, “Grotesque: Noisy group testing (quick and efficient),” *51st Annual Conference on Communication, Control, and Computing (Allerton)*, 2013.

[23] R. Pedarsani, K. Lee, and K. Ramchandran, “Phasecode: Fast and efficient compressive phase retrieval based on sparse-graph-codes,” *IEEE Transactions on Information Theory*, 2014.

[24] I.M.Stancu-Minasian, *Fractional Programming Theory,Methods and Applications*. Springer, 1997.

[25] S. Kudekar, T. J. Richardson, and R. L. Urbanke, “Threshold saturation via spatial coupling: Why convolutional ldpc ensembles perform so well over the bec,” *IEEE Transactions on Information Theory*, vol. 57, 2011.

[26] P. Damaschke, “Threshold group testing,” *General Theory of Information Transfer and Combinatorics*, 2006.

[27] C. L. Chan, S. Cai, M. Bakshi, S. Jaggi, and V. Saligrama, “Stochastic threshold group testing,” *2013 IEEE Information Theory Workshop (ITW)*, 2013.

[28] C. Verdun, “Compressed sensing,” Master’s thesis, Federal University of Rio de Janeiro (UFRJ), 2016.

A Appendix

a Computer Science Notation

In this section we present a formal definition of the $\mathcal{O}$ and Ω notation, just to clarify possible doubts for readers that are not familiar with the notation.

Let f and g be two functions. We say that $f(x)$ = $\mathcal{O}(g(x))$ if there is a constant $C > 0$ such that:

$$|f(x)| \leq |g(x)| \; \forall x \in Domain(f) \cap Domain(g)$$

For lower bounds, we say that $f(x) = \Omega(g(x))$ if there is a constant $C > 0$ such that:

$$|f(x)| \geq C|g(x)|$$

We shall observe that $f(x) = \mathcal{O}(g(x))$ if and only if $g(x) = \Omega(f(x))$

b Numerical Linear Algebra & Optimization

We present some basic definitions about linear algebra and the inserction between sparsity and optimization.

Definition 15. Let $\mathbb{E}$ be a vector space over the real numbers, we say that $\mathbb{E}$ is a normed vector space if there is a function : $||.|| : \mathbb{E} \rightarrow \mathbb{R}$ such that:

- $||x|| = 0 \iff x = 0$
- $||\lambda x|| = |\lambda| \; ||x|| \; \forall \lambda \in \mathbb{R} \; \forall x \in \mathbb{E}$
- $\forall x, y \in \mathbb{E} \; ||x + y|| \leq ||x|| + ||y||$

Example 8. Let $\mathbb{E} = \mathbb{R}^n$, the norm l^p is given by $(\sum_{i=1}^{n} |\mathbf{x}|^p)^{1/p}$

Remark 14. We left to the readers to check that the norm l^p is definitely a norm

An important concept is the sparsity of a vector due to its provide a low computational cost for many operations and algorithms. A vector is said to be sparse with has only a few non-zero entries, this definition is inaccurate but gives us the idea of what we are searching for. For a more formal definition, we define a s-sparse vector.

Definition 16. A vector $\mathbf{x} \in \mathbb{R}^n$ is said to be s-sparse if there are at maximum s non-zero entries

It is a well known fact from linear algebra that if the system $\mathbf{y} = \mathbf{A}\mathbf{x}$ is solvable and $\mathbf{A}$ has low rank then we have infinite number of solutions. Then its natural to consider an optimization problem that finds the most sparse solution for this system. It is easy to check that $||.||_0$ defined to be the number of non-zero entries of a vector, i.e, how sparse the vector is, is not a norm (The second condition fails). This fact complicates the optimization approach in the section 1, the problem turns to be NP-Hard, i,e, no hope to find a fast solution to the problem

A common approach to relax the problem is to minimize the norm l^1. It is a miracle from mathematics that l^1 promotes sparsity, while the other l norms do not. The geometrical intuition for this fact is that l^p norms are norms, so its obey the third condition mentioned in the above definition that gives us convexity and for $p \leq 1$ the quasi norm promotes sparsity, therefore the l^1 is the unique norm that promotes sparsity.

In the past decades the search for a perfect algorithm to the optimization problems in the 1 was intense. We refer [28] to very detailed material about it.

C Miscellaneous

From calculus, D'Alambert criterion for series is the following theorem

Theorem 10. *Let $a_n \neq 0$ a sequence of real numbers. If exists a constant c such that $\frac{|a_{n+1}|}{|a_n|} \leq c < 1$ for every n sufficient large, then the infinite series $\sum a_n$ converge absolutely.*

The proof is simple and can be found in any book about advanced calculus or basic analysis

From probability theory the Poisson process characterizes the arrival process, in mathematical terms we write :

$$P(X = k) = \frac{\lambda^k exp(-\lambda)}{k!} \tag{18}$$

Where X is a random variable and $\lambda > 0$ is the Poisson parameter

From the abstract algebra theory, the formal definition of a algebraic field is given below:

Definition 17. A field $\mathbb{F}$ is set equipped with two operations, addition(+) and multiplication(.). These properties have the following properties:

- Associativity
- Commutativity
- Inverses, except for 0
- Distributivity

Example 9. The set of real numbers $\mathbb{R}$ with the common addition and multiplication. A non trivial example is the binary field $\mathbb{F}_2$ with $\oplus$ and common multiplication. The addition $\oplus$ is defined to be operation mod-2 :

- $0 \oplus 0 = 0$
- $0 \oplus 1 = 1$
- $1 \oplus 0 = 0$
- $1 \oplus 1 = 1$

We can observe that this operation is equivalent to the logical operator XOR.