# This document is downloaded from DR-NTU (https://dr.ntu.edu.sg) Nanyang Technological University, Singapore. 

## On index coding with side information

Dau, Son Hoang

2012

Dau, S. H. (2012). On index coding with side information. Doctoral thesis, Nanyang Technological University, Singapore.
https://hdl.handle.net/10356/48642
https://doi.org/10.32657/10356/48642


## ON INDEX CODING WITH SIDE INFORMATION

DAU SON HOANG

SCHOOL OF PHYSICAL \& MATHEMATICAL SCIENCES

# ON INDEX CODING WITH SIDE INFORMATION 

## DAU SON HOANG

## School of Physical \& Mathematical Sciences

A thesis submitted to the Nanyang Technological University in partial fulfillment of the requirement for the degree of Doctor of Philosophy

## ACKNOWLEDGEMENTS

This thesis would not have been possible without the help and support of the very kind people.

First and foremost, I would like to express my sincerest gratitude to my supervisors, Professor Chee Yeow Meng and Professor Ling San, for their priceless guidance and support, in both personal aspect and academic aspect. I am extremely grateful for their patience, kindness, and attentiveness towards me during my time at NTU.

I would also like to give a special thank to Vitaly Skachek, who worked very closely to me during my time as a PhD student. It was him who first introduced the Index Coding with Side Information problem to me. Many of the questions, ideas, and suggestions he put to me have actually been turned into important theorems in this thesis. It was the fruitful discussions with Vitaly and Yeow Meng that shaped the main content of the thesis. Vitaly's generous supports and encouragements have been invaluable to me.

I am indebted to many of my friends for their helps with computer related issues, among whom are Birk Bernd Eisermann, Nguyen Luong Dong, Nguyen Quoc Chinh, Punarbasu Purkayastha, Radoslav Kirov, Tran Duc Hieu, and Wei Lei. I also thank Melvin for his help with the use of SPMS servers.

Finally, I thank my Mom and my Dad for their endless encouragements and supports. Their love makes me feel more confident and stronger.


#### Abstract

Building communication schemes which allow participants to communicate efficiently has always been a challenging yet intriguing problem for information theorists. Index Coding with Side Information (ICSI), first introduced by Birk and Kol (1998) as Coding on Demand by an Informed Source (ISCOD), is a communication scheme dealing with a broadcast channel in which receivers have prior side information about the messages to be broadcast. Exploiting the knowledge about the side information, the sender may significantly reduce the number of required transmissions compared with the naive approach. As a consequence, the efficiency of the communication over this type of broadcast channels could be dramatically improved. Apart from being a special case of the well-known (non-multicast) Network Coding problem (Alhswede et al. (2000), Koetter and Médard (2003)), the ICSI problem has also found various potential applications on its own, such as audioand video-on-demand, daily newspaper delivery, data pushing, and oppoturnistic wireless networks.

While most of the known works on ICSI focus on the performances of index codes of various kinds, the security and error correction aspects of those codes have never been examined. Although the issues of security and error correction have been thoroughly investigated in multicast Network Coding, on the side of non-multicast Network Coding, where ICSI is one example, those issues are much less understood. In this work, we make an attempt to fill in this gap.


We first analyze the block security level of the linear index code based on a matrix $\boldsymbol{L}$, by examining the minimum distance and the dual distance of the column space $\mathcal{C}(\boldsymbol{L})$ of $\boldsymbol{L}$. Here the block security is a generalization of the notion of weak security, introduced by Bhattad and Narayanan (2005) in the setups of Network Coding. It turns out that these two distances of $\mathcal{C}(\boldsymbol{L})$ specify two closely related thresholds, one is for the code to be block secure, the other is for it to be completely insecure. When $\mathcal{C}(\boldsymbol{L})$ is an MDS code, the two thresholds are actually tight. We then proceed to examine the strong security of linear index codes, which is the information-theoretic security in the context of ICSI. We show that the coset coding technique, an indispensable tool in securing a network code in the setting of Network Coding, also provides us with an optimal solution for securing an index code.

Subsequently, we examine the index coding schemes in which errors are involved. Several bounds and constructions for linear error-correcting index codes are established. Syndrome decoding for error-correcting index codes, which is somewhat different in nature from that for classical error-correcting codes, is also investigated. Furthermore, we study the so-called static error-correcting index codes. Each of such codes works for a family of instances of the ICSI problem. Analogous bounds and constructions for nonlinear error-correcting index codes, as for linear case, are also developed.

The second line of work in this thesis is on the computational aspects of the ICSI problem. As shown in the work of Bar-Yossef et al. (2006), the minimum number of transmissions required by a linear index code is equal to the so-called minrank of the corresponding side information digraph. However, as shown by Peeters (1996), finding the minrank of a side information graph is an NP-hard problem. There are very few known families of (di)graphs for which minranks can be found in polynomial time. They are perfect graphs, odd holes (cycles) and odd anti-holes, outerplanar
graphs, and acyclic digraphs. In this work, we discover several more such families, including connectively reducible digraphs, line digraphs of partially planar digraphs, and graphs possessing a special tree structure. We also characterize (di)graphs of extreme minranks, that is, (di)graphs with minranks close to one or to their orders (i.e., number of vertices). Based on one of these characterizations, we are able to show that deciding whether a digraph has minrank two is an NP-complete problem. For comparison, the same decision problem for graphs can be solved in polynomial time. Finally, based on the work of Chaudhry and Sprintson (2008), we write a computer program to find the minranks of all non-isomorphic graphs of orders up to 10 .

## CONTENTS

1. Introduction ..... 1
1.1 Background ..... 1
1.2 Contributions of this Thesis ..... 3
1.3 Organization of this Thesis ..... 12
1.4 Notation and Definitions ..... 12
1.4.1 Coding Theory Terminology ..... 12
1.4.2 Linear Algebra Terminology ..... 14
1.4.3 Graph Theory Terminology ..... 15
1.4.4 Information Theory Terminology ..... 19
2. Secure Index Code with Side Information ..... 21
2.1 Index Coding and Some Basic Results ..... 22
2.2 Block Secure Linear Index Codes ..... 26
2.2.1 Block Security and Weak Security ..... 26
2.2.2 Necessary and Sufficient Conditions for Block Security ..... 28
2.2.3 Complete Insecurity ..... 38
2.3 Index Coding with Side and Restricted Information ..... 42
2.4 Strongly Secure Index Code with Side Information ..... 43
2.4.1 A Lower Bound on the Length of a Strongly Secure Index Code ..... 44
2.4.2 A Construction of Optimal Strongly Secure Index Codes ..... 50
3. Index Code and Error Correction ..... 54
3.1 Error-Correcting Index Code with Side Information ..... 55
3.2 The $\alpha$-Bound and the $\kappa$-Bound ..... 61
3.3 The Singleton Bound ..... 69
3.4 Random codes ..... 71
3.5 Strongly Secure Linear Error-Correcting Index Codes ..... 73
3.5.1 A Lower Bound on the Length of a Strongly Secure Linear Error-Correcting Index Code ..... 74
3.5.2 A Construction of an Optimal Strongly Secure Linear Error- Correcting Index Code ..... 75
3.6 Syndrome Decoding ..... 77
3.7 Static Error-Correcting Index Codes ..... 82
3.7.1 Static Error-Correcting Index Codes ..... 82
3.7.2 An Application: Weakly Resilient Functions ..... 84
3.7.3 Bounds and Constructions ..... 87
3.8 Nonlinear Error-Correcting Index Codes ..... 92
4. (Di)Graphs of Extreme MinRanks ..... 97
4.1 (Strongly) Connected Components and MinRanks ..... 97
4.2 (Di)Graphs of Extreme MinRanks ..... 100
4.2.1 (Di)Graphs of MinRank One ..... 100
4.2.2 (Di)Graphs of MinRank Two ..... 100
4.2.3 (Di)Graphs of MinRanks Equal to Their Orders ..... 103
4.2.4 Graphs of MinRanks One Less Than Their Orders ..... 105
5. Computation of MinRanks of Side Information (Di)Graphs ..... 109
5.1 The Hardness of the MinRank Problem for Digraphs ..... 109
5.2 Circuit-Packing Bound ..... 114
5.2.1 The Bound ..... 114
5.2.2 Digraphs Attaining Circuit-Packing Bound ..... 118
5.3 On MinRanks of Graphs Having Tree Structures of Type I ..... 127
5.3.1 Tree Structure of Type I ..... 127
5.3.2 An Algorithm for MinRanks of Graphs in $\mathscr{F}_{1}(c)$ ..... 129
5.3.3 An Algorithm to Recognize a Graph in $\mathscr{F}_{2}(c)$ ..... 141
5.4 MinRanks of (Di)Graphs of Small Orders ..... 145
6. Conclusion ..... 147
Bibliography ..... 151
Appendix ..... 159
A. An Exact Algorithm for Finding MinRanks of (Di)Graphs over $\mathbb{F}_{2}$ ..... 160
A. 1 The Algorithm ..... 160
A. 2 Reduction to SAT Problem ..... 163
A. 3 Generating Lists of Non-Isomorphic Graphs ..... 169
B. MinRanks of Graphs of Small Orders ..... 171
B. 1 MinRanks of Graphs of Orders One to Four ..... 171
B. 2 MinRanks of Graphs of Order Five ..... 172
B. 3 MinRanks of Graphs of Order Six ..... 173
Index ..... 182

## 1. INTRODUCTION

### 1.1 Background

The problem of Index Coding with Side Information (ICSI) was introduced by Birk and Kol $[10,11]$. The coding scheme studied in the ICSI problem was originally called Coding on Demand by an Informed Source (ISCOD). It was motivated by applications such as audio and video-on-demand, and daily newspaper delivery, in which a server (sender) broadcasts a set of messages to a set of clients (receivers). During the transmission, each client might miss a certain part of the data, due to intermittent reception, limited storage capacity or any other reasons. The server has to find a way to deliver to each client all the missing messages, yet spending a minimum number of transmissions. Via a slow backward channel, the clients let the server know which messages they already have in their possession, and which messages they request. By exploiting this information, the amount of the overall transmissions can be significantly reduced [11].

Another possible application of the ICSI problem is in opportunistic wireless networks. These are the networks in which a wireless node can opportunistically listen to the wireless channel. As a result, a node may obtain packets that were not designated to it $[28,41,42]$. In this way, a node obtains some side information about the transmitted data. Exploiting this additional knowledge may help to increase the throughput of the system.

Consider the toy example in Fig. 1.1. It presents a scenario with one sender and four receivers. Each receiver requires a different information packet (or message). The naive approach requires four separate transmissions, one transmission per an information packet. However, by utilizing the knowledge about the subsets of messages that receivers already have, and by coding the transmitted data, the sender can satisfy all the demands by broadcasting just one coded packet.

The ICSI problem has been the subject of several recent studies $[2,3,8,15,16,21$, $22,25,27,28,36,46,69]$. On the one hand, this problem is a special case of the wellknown (non-multicast) network coding (NC) problem [1,43]. On the other hand, it was shown that every instance of the NC problem can be reduced to an instance of the ICSI problem in the following sense. For each NC instance, we can construct an ICSI instance such that there exists a scalar linear network code for the NC instance if and only if there exists a perfect scalar linear index code for the corresponding ICSI instance (see [27,28] for more details).


Fig. 1.1: An example of the ICSI problem

Several previous works focused on the design of efficient index codes. Given an
instance of the ICSI problem, Bar-Yossef et al. [3, 4] proved that finding the best binary scalar linear index code is equivalent to finding the so-called minrank of a (di)graph (together with a matrix whose rank is equal to the minrank), which is an NP-hard problem $[3,53]$. Here scalar linear index codes refer to linear index codes in which each message is a symbol in the underlying field $\mathbb{F}_{q}$. By contrast, in vector linear index codes each message is a vector over $\mathbb{F}_{q}$. Lubetzky and Stav [46] showed that there exist instances for which scalar linear index codes over nonbinary fields and combinations of linear index codes over two different fields outperform binary scalar linear index codes. El Rouayheb et al. [27, 28] showed that for certain instances of the ICSI problem, vector linear index codes achieve strictly higher transmission rate than scalar linear index codes. They also pointed out that there exist instances in which vector nonlinear index codes outperform vector linear index codes. Vector nonlinear index codes were also shown to outperform scalar nonlinear index codes for certain instances by Alon et al. [2]. Several heuristic solutions for the ICSI problem were proposed by El Rouayheb et al. [25] and by Chaudhry and Sprintson [16]. Optimal scalar linear index codes for an ICSI instance described by a random side information (di)graph was investigated by Haviv and Langberg [36]. A dynamic programming approach (Bellman [5]) was established by Berliner and Langberg [8] to find in polynomial time optimal scalar linear index codes for ICSI instances described by outerplanar graphs.

### 1.2 Contributions of this Thesis

Security issues and computational issues of the index coding schemes are the main topics to be discussed in this thesis.

In Chapter 2, we study the security aspects of linear index codes. This is the
first known work in this direction. We restrict ourselves to scalar linear index codes due to the following reasons. First, although nonlinear codes are theoretically interesting, they are usually not practically attractive, due to the high complexity in encoding and decoding. It is known that vector linear index codes can achieve better transmission rates than their scalar counterparts for certain instances of the ICSI problem [27, 28]. However, if the block length of a vector index code for an instance is fixed, we can model this vector index code as a scalar index code for another instance of the ICSI problem (see the proof of Proposition 5.2.16).

A linear index code maps $\boldsymbol{x} \in \mathbb{F}_{q}^{n}$ onto $\boldsymbol{x} \boldsymbol{L}$, where $\boldsymbol{L}$ is an $n \times N$ matrix over $\mathbb{F}_{q}$, and $n, N \in \mathbb{N}$. Here $\mathbb{F}_{q}$ denotes the finite field with $q$ elements. The number of transmissions required when such an index code is used is $N$. We call $N$ the length of this index code.

We show that each linear index code provides a certain level of information security. We describe this result in more details as follows. Let the code $\mathcal{C}(\boldsymbol{L})$ be spanned by the columns of $\boldsymbol{L}$, and let $d$ and $d^{\perp}$ be its minimum distance and dual distance, respectively. Assume that there exists an adversary who can listen to all transmissions from the sender. We say that a particular adversary is of strength $t$ if it has $t$ messages in its possession. An index code is said to be $b$-block secure against all adversaries of strength $t$ if every adversary of strength $t$ has no information (in Shannon's sense) about any block of $b$ messages that are not in his possession. In contrast, an index code is said to be completely insecure against any adversary of strength $t$ if an adversary who possesses an arbitrary set of $t$ messages is always able to reconstruct all of the other $n-t$ messages. Then, we show that the index code based on $\boldsymbol{L}$ is $(d-1-t)$-block secure against all adversaries of strength $t \leq d-2$ and is completely insecure against any adversary of strength at least $n-d^{\perp}+$ 1 (Theorems 2.2.9 and 2.2.12). If $\mathcal{C}(\boldsymbol{L})$ is an MDS code, then the two bounds
coincide. The technique used in the proof for this result is reminiscent of that used in the constructions of (multiple) secret sharing schemes from linear error-correcting codes $[23,50]$. The results on the security of linear index codes can be further employed to analyze the existence of solutions for a natural generalization of the ICSI problem, so-called the Index Coding with Side and Restricted Information (ICSRI) problem. In that problem, it is required that some receivers have no information about some messages.

In the sequel, we also consider linear randomized index codes, which are based on the use of random symbols. Using such index codes, the sender first generates random symbols, mixes them with the original packets, and then broadcasts the mixed packets. We show that the coset coding technique (which has been successfully employed in Secure Network Coding literature [12, 26, 29, 59, 71]) yields an optimal strongly secure linear randomized index code of length $\kappa_{q}+\mu$ (Theorems 2.4.10). This randomized index code is strongly secure against an adversary that
(i) has $t$ arbitrary messages in advance;
(ii) eavesdrops at most $\mu$ transmissions.

In other words, such an adversary gains no information (in Shannon's sense) about the messages that he does not possess. Observe that the length of the optimal code does not depend on $t$. In fact, the construction of such a code is independent of $t$. Here $\kappa_{q}$ denotes the minrank over $\mathbb{F}_{q}$ of the side information hypergraph that corresponds to the instance of the ICSI problem. As shown later in Corollary 2.2.8, $\kappa_{q}$ is equal to the length of an optimal scalar linear index code over $\mathbb{F}_{q}$.

It is worth mentioning that the security models considered in this work (block security and strong security) are special cases of a more general model referred to as security against guessing in the context of Network Coding (see Bhattad and

Narayanan [9]). In their model, an adversary can obtain some set $M$ of linear combinations of the (source) messages by eavesdropping on a certain set of links in the communication network that connects the sender and the receivers. Let $\left\{G_{p}, U_{p}\right\}$ ( $p \in P$ ) be pairs of sets of messages. The adversary can guess perfectly all messages in $G_{p}(p \in P)$. In other words, the adversary has some side information about the messages. The security requirement is that the adversary has no information (in Shannon's sense) about $U_{p}$ given his knowledge of $M$ (eavesdropped information) and $G_{p}$ (side information) for every $p \in P$. When $G_{p}$ is any subset of $t$ messages and $U_{p}$ is any subset of $b$ messages $\left(G_{p} \cap U_{p}=\varnothing\right)$, this model reduces to the $b$ block security model. When $G_{p}$ is any subset of $t$ messages and $U_{p}$ is the set of the remaining $n-t$ messages, the model reduces to the strong security model.

Most of previous works on the security aspects (and on the error-correction aspect, as a special case) of network coding dealt with the multicast scenario. One of the main reasons for this limitation is that the optimal simultaneous transmission rates for non-multicast networks have not been well characterized yet. It is known that the ICSI problem can be modeled as a special case of the non-multicast Network Coding problem [2,28]. This fact restricts the ability to derive the results on the security of the index coding schemes from the existing results on the security of network coding schemes.

The preceding works on the ICSI problem consider a scenario where the transmissions are error-free. In practice, of course, this might not be the case. In Chapter 3, we assume that the transmitted symbols are subject to errors and extend some known results on index coding to a case where each receiver can correct up to a certain number of errors. It turns out that the problem of designing such errorcorrecting index codes (ECIC's) naturally generalizes the problem of constructing classical error-correcting codes.

More specifically, assume that the number of messages that the sender possesses is $n$, and that the maximum number of errors is $\delta$. We show that the problem of constructing an ECIC of minimum length is equivalent to the problem of constructing a matrix $\boldsymbol{L}$ that has $n$ rows and minimum number of columns, such that

$$
\text { wt }(\boldsymbol{z} \boldsymbol{L}) \geq 2 \delta+1 \text { for all } \boldsymbol{z} \in \mathcal{I}
$$

where $\mathcal{I}$ is a certain subset of $\mathbb{F}_{q}^{n} \backslash\{\mathbf{0}\}$ (Lemma 3.1.3). Here $w t(\boldsymbol{x})$ denotes the Hamming weight of the vector $\boldsymbol{x}$ and $\mathbf{0}$ is the all-zeros vector. If $\mathcal{I}=\mathbb{F}_{q}^{n} \backslash\{\mathbf{0}\}$, this problem becomes equivalent to the well-known problem of designing a shortestlength linear error-correcting code of given dimension and minimum distance.

We establish a lower bound (the $\alpha$-bound) and an upper bound (the $\kappa$-bound) on the shortest length of a linear ECIC that is able to correct every error pattern of size up to $\delta$ (Theorem 3.2.5 and Proposition 3.2.8). These bounds are described in more details as follows. Let $\mathcal{H}$ be the side information hypergraph that describes an instance of the ICSI problem. Let $\mathcal{N}_{q}[\delta, \mathcal{H}]$ denote the minimum length of a linear ECIC over $\mathbb{F}_{q}$ which ensures that every receiver $R_{i}$ can recover the desired message if the number of errors is at most $\delta$. We use notation $N_{q}[k, d]$ for the minimum length of a linear error-correcting code of dimension $k$ and minimum distance $d$ over $\mathbb{F}_{q}$. The $\alpha$-bound and the $\kappa$-bound are the following

$$
\begin{equation*}
N_{q}[\alpha(\mathcal{H}), 2 \delta+1] \leq \mathcal{N}_{q}[\delta, \mathcal{H}] \leq N_{q}\left[\kappa_{q}(\mathcal{H}), 2 \delta+1\right] \tag{1.1}
\end{equation*}
$$

where $\alpha(\mathcal{H})$ is the generalized independence number and $\kappa_{q}(\mathcal{H})$ is the minrank (over $\mathbb{F}_{q}$ ) of $\mathcal{H}$. The $\kappa$-bound (the second inequality) is obtained simply by concatenating an optimal linear traditional error-correcting code and an optimal linear index code.

For linear index codes, we also derive an analog of the Singleton bound (Theorem 3.3.1). This result implies that (over sufficiently large alphabets) the concatenation of a standard MDS error-correcting code with an optimal linear index code yields an optimal linear error-correcting index code. Finally, we consider a random ECIC. By analyzing its parameters, we obtain an upper bound on its length (Theorem 3.4.1).

When the side information hypergraph is a pentagon, and $\delta=2$, the inequalities in (1.1) are shown to be strict. This implies that a concatenated scheme based on a classical error-correcting code and on a linear non-error-correcting index code does not necessarily yield an optimal linear error-correcting index code. Since the ICSI problem can also be viewed as a source coding problem [2, 3], this example demonstrates that sometimes designing a single code for both source and channel coding can result in a smaller number of transmissions.

Based on the results obtained in Chapter 2 and the results on ECIC's in this chapter (Chapter 3), we give a construction of an optimal strongly secure linear randomized error-correcting index code of length $\kappa_{q}+\mu+2 \delta$ (Theorem 3.5.5). This randomized index code is strongly secure against an adversary that
(i) has $t$ arbitrary messages in advance;
(ii) eavesdrops at most $\mu$ transmissions;
(iii) corrupts at most $\delta$ transmissions.

The decoding of a linear ECIC is somewhat different from that of a classical errorcorrecting code. There is no longer a need for a complete recovery of the whole information vector. As a consequence, each receiver no longer has to determine the exact error vector in order to decode the desired message. We analyze the
decoding criteria for the ECIC's and show that the syndrome decoding, which might be different for each receiver, outputs the correct result, provided that the number of errors does not exceed the error-correcting capability of the code.

An ECIC is called static under a family of instances of the ICSI problem if it works for all of these instances. Such an ECIC is desirable since it remains relevant as long as the parameters of the problem vary within a particular range. Bounds and constructions for static ECIC's are studied in Section 3.7. Connections between static ECIC's and weakly resilient vectorial Boolean functions are also investigated.

At the end of Chapter 3, in Section 3.8, we briefly discuss the nonlinear ECIC's and establish several analogous results for bounds on minimum lengths of nonlinear ECIC's.

As shown by Bar-Yossef et al. [3, 4], the minrank of a digraph is precisely the number of transmissions required in an optimal scalar linear index code for an instance of the ICSI problem described by that digraph. The concept of minrank of a graph was first introduced by Haemers [34], which serves as an upper bound for the celebrated Shannon capacity of a graph [57]. This upper bound, as pointed out by Haemers himself, although is usually not as good as the Lovász bound [45], is sometimes tighter and easier to compute. Unfortunately, as shown by Peeters [53], computing the minrank of a general graph (that is, the MinRank problem) is a hard task. More specifically, Peeters showed that deciding whether a graph has minrank three is an NP-complete problem. The interest on the MinRank problem was resumed after the work of Bar-Yossef et al. [3], which proved that the binary minrank of a digraph $\mathcal{D}$ characterizes the optimal binary scalar linear index code for an ICSI instance with side information digraph $\mathcal{D}$. Exact and heuristic algorithms to find the minrank over $\mathbb{F}_{2}$ of a hypergraph (and a (di)graph as a special case) were developed in the work of Chaudhry and Sprintson [16]. The minranks of random
(di)graphs are investigated by Haviv and Langberg [36]. A dynamic programming approach was proposed by Berliner and Langberg [8] to compute in polynomial time the minranks of outerplanar graphs. It is also worth noting that approximating minranks of graphs within any constant ratio is known to be NP-hard (see Langberg and Sprintson [44]).

In Chapter 4, we make an attempt to characterize the (di)graphs that have extreme minranks. In particular, it is shown that a digraph has minrank two over $\mathbb{F}_{2}$ if and only if $\mathcal{D}$ is not a complete digraph and its complement, $\overline{\mathcal{D}}$, is fairly 3 -colorable (Corollary 4.2.5). Here a digraph is fairly 3 -colorable if one can color its vertices by three colors in such a way that not only the endpoints of the same arc have different colors but also all out-neighbors of the same vertex must share the same color. Based on this characterization, we show later in Chapter 5 that the problem of deciding whether a digraph has minrank two is NP-complete. In contrast, for the case of graphs, it is known that the minrank of a graph $\mathcal{G}$ is two if and only if $\mathcal{G}$ is not a complete graph and its complement, $\overline{\mathcal{G}}$, is bipartite, a condition which can be verified in polynomial time (see, for instance West [67, p. 495]). Note that a graph is bipartite if and only if it is 2-colorable.

The other characterizations are the following. A graph has minrank equal to its order if and only if it contains no edges. By contrast, a digraph has minrank equal to its order if and only if it contains no circuits. A graph of order $n$ has minrank $n-1$ if and only if its largest connected component is a star graph and the other components are one-vertex graphs.

In Chapter 5, we focus on the computational aspects of the ICSI problem. We first establish that deciding whether a digraph has minrank two is already an NPcomplete problem (Corollary 5.1.2). Afterwards, a new upper bound on the minrank of a digraph, so-called the circuit-packing bound, is introduced (Proposition 5.2.3).

In certain cases, the circuit-packing bound is shown to be far better than the cliquecover bound. So far, families of (di)graphs whose minranks are either known or computable in polynomial time are the following. For graphs, they are odd holes and odd anti-holes, perfect graphs, and outerplanar graphs. For digraphs, they are acyclic digraphs. The circuit-packing bound, together with the lower bound based on the order of a maximum acyclic subgraph, enables us to derive several new families of digraphs whose minranks are computable in polynomial time. They are the families of connectively reducible digraphs and line digraphs of partially planar digraphs. For those two families of digraphs, the circuit-packing bound is tight. This bound is also tight for other families of digraphs, including digraphs that pack (a digraph packs if all of its subgraphs satisfy the min-max vertex equality, see Section 5.2.2 for more details), line digraphs of fully reducible flow digraphs, and line digraphs of special Eulerian digraphs. Moreover, for ICSI instances described by the aforementioned families of digraphs, scalar linear index codes are shown to be optimal (Proposition 5.2.16).

Based on the fact that outerplanar graphs can be decomposed into tree structures that support a dynamic programming approach, Berliner and Langberg [8] established an efficient algorithm to find minranks of outerplanar graphs. Inspired by their work, at the end of Chapter 5, we develop a dynamic programming algorithm to compute in polynomial time the minranks of graphs having a special structure. Such a graph can be described as a compound rooted tree, the nodes of which are graphs whose minranks can be computed in polynomial time. The algorithm computes the minranks of the subtrees from the leaves to the root, in a bottom-up manner. The task of computing the minrank of the graph is accomplished when the computation reaches the root of the compound tree. We end Chapter 5 by mentioning that the minranks of all non-isomorphic graphs of order up to 10
can be found using a computer program that combines a SAT-based approach and a Branch-and-Bound approach.

### 1.3 Organization of this Thesis

The main content of the thesis is organized into six chapters.
Chapter 1 provides a background on the ICSI problem, which is the main subject of this work. It also summarizes the key results obtained in the thesis and contains most of the basic definitions and notation used throughout the thesis.

Chapters 2 and 3 discuss the security aspects of scalar linear index codes. Bounds and constructions for error-correcting index codes and strongly secure index codes are established. A decoding algorithm for error-correcting index codes is proposed. Most of the results in these two chapters contain the minrank of a side information hypergraph (and (di)graph as a special case) in their formulations.

A deeper investigation of the minranks of side information (di)graphs is carried out in Chapters 4 and 5 . The structures of (di)graphs having extreme minranks are examined. New families of (di)graphs whose minranks can be found in polynomial time are discovered. Details on the computation of minranks of graphs of small orders can be found in the Appendix.

The thesis is concluded in Chapter 6. Several intriguing open problems for future research are proposed.

### 1.4 Notation and Definitions

### 1.4.1 Coding Theory Terminology

We use $\mathbb{F}_{q}$ to denote the finite field of $q$ elements, where $q$ is a power of prime. We denote by $\mathbb{F}_{q}^{*}$ the set of all nonzero elements of $\mathbb{F}_{q}$.

Let $[n]=\{1,2, \ldots, n\}$. For the vectors $\boldsymbol{u}=\left(u_{1}, u_{2}, \ldots, u_{n}\right) \in \mathbb{F}_{q}^{n}$ and $\boldsymbol{v}=$ $\left(v_{1}, v_{2}, \ldots, v_{n}\right) \in \mathbb{F}_{q}^{n}$, the (Hamming) distance between $\boldsymbol{u}$ and $\boldsymbol{v}$ is defined to be the number of coordinates where $\boldsymbol{u}$ and $\boldsymbol{v}$ differ, namely,

$$
\mathrm{d}(\boldsymbol{u}, \boldsymbol{v})=\left|\left\{i \in[n]: u_{i} \neq v_{i}\right\}\right| .
$$

If $\boldsymbol{u} \in \mathbb{F}_{q}^{n}$ and $\boldsymbol{M} \subseteq \mathbb{F}_{q}^{n}$ is a set of vectors (for instance, a vector subspace), then the last definition can be extended to

$$
\mathrm{d}(\boldsymbol{u}, \boldsymbol{M})=\min _{\boldsymbol{v} \in M} \mathrm{~d}(\boldsymbol{u}, \boldsymbol{v}) .
$$

The support of a vector $\boldsymbol{u}=\left(u_{1}, u_{2}, \ldots, u_{n}\right) \in \mathbb{F}_{q}^{n}$ is defined to be the set

$$
\operatorname{supp}(\boldsymbol{u})=\left\{i \in[n]: u_{i} \neq 0\right\} .
$$

The (Hamming) weight of a vector $\boldsymbol{u}$, denoted $w t(\boldsymbol{u})$, is defined to be $|\operatorname{supp}(\boldsymbol{u})|$, the number of nonzero coordinates of $\boldsymbol{u}$. Suppose $E \subseteq[n]$. We write $\boldsymbol{u} \triangleleft E$ whenever $\operatorname{supp}(\boldsymbol{u}) \subseteq E$.

A $k$-dimensional subspace $\mathscr{C}$ of $\mathbb{F}_{q}^{n}$ is called a linear $[n, k, d]_{q}$ code if the minimum distance of $\mathscr{C}$,

$$
\mathrm{d}(\mathscr{C}) \triangleq \min _{\boldsymbol{u} \in \mathscr{C}, \boldsymbol{v} \in \mathscr{C}, \boldsymbol{u} \neq \boldsymbol{v}} \mathrm{d}(\boldsymbol{u}, \boldsymbol{v})
$$

is equal to $d$. Sometimes we may use the notation $[n, k]_{q}$ for the sake of simplicity. We call $n$ the length and $k$ the dimension of the code. The vectors in $\mathscr{C}$ are called codewords. It is easy to see that the minimum weight of a nonzero codeword in a linear code $\mathscr{C}$ is equal to its minimum distance $\mathrm{d}(\mathscr{C})$. A generator matrix $\boldsymbol{G}$ of an $[n, k]_{q}$ code $\mathscr{C}$ is a $k \times n$ matrix whose rows are linearly independent codewords of
$\mathscr{C}$. Then $\mathscr{C}=\left\{\boldsymbol{y} \boldsymbol{G}: \boldsymbol{y} \in \mathbb{F}_{q}^{k}\right\}$. The parity-check matrix of $\mathscr{C}$ is an $(n-k) \times n$ matrix $\boldsymbol{H}$ over $\mathbb{F}_{q}$ such that $\boldsymbol{c} \in \mathscr{C} \Leftrightarrow \boldsymbol{H} \boldsymbol{c}^{T}=\mathbf{0}^{T}$. Given $q, k$, and $d$, let $N_{q}[k, d]$ denote the length of the shortest linear code over $\mathbb{F}_{q}$ that has dimension $k$ and minimum distance $d$.

The dual code or dual space of $\mathscr{C}$ is defined as $\mathscr{C}^{\perp}=\left\{\boldsymbol{u} \in \mathbb{F}_{q}^{n}: \boldsymbol{u c}^{T}=\right.$ 0 for all $\boldsymbol{c} \in \mathscr{C}\}$. The minimum distance of $\mathscr{C}^{\perp}$, namely $\mathrm{d}\left(\mathscr{C}^{\perp}\right)$, is called the dual distance of $\mathscr{C}$.

An $(n, M, d)_{q}$ code is an $M$-subset $\mathscr{C}$ of $\mathbb{F}_{q}^{n}$ that satisfies the property that the (Hamming) distance between every two distinct vectors of $\mathscr{C}$ is at least $d$. We call $M$ the size of the code. We, again, call $n$ and $d$ the length and the distance of the code, respectively. Given $q, M$, and $d$, let $N_{q}(M, d)$ denote the length of a shortest code of size $M$ and distance $d$.

The following upper bound on the minimum distance of a $q$-ary linear code is well known [49, Chapter 1].

Theorem 1.4.1 (Singleton bound). For an $[n, k, d]_{q}$ code, we have $d \leq n-k+1$.

Codes attaining this bound are called maximum distance separable (MDS) codes.

### 1.4.2 Linear Algebra Terminology

We use

$$
\boldsymbol{e}_{i}=(\underbrace{0, \ldots, 0}_{i-1}, 1, \underbrace{0, \ldots, 0}_{n-i}) \in \mathbb{F}_{q}^{n}
$$

to denote the unit vector, which has a one at the $i$ th position, and zeros elsewhere. For a vector $\boldsymbol{y}=\left(y_{1}, y_{2}, \ldots, y_{n}\right)$ and a subset $B=\left\{i_{1}, i_{2}, \ldots, i_{b}\right\}$ of $[n]$, where $i_{1}<i_{2}<\cdots<i_{b}$, let $\boldsymbol{y}_{B}$ denote the vector $\left(y_{i_{1}}, y_{i_{2}}, \ldots, y_{i_{b}}\right)$.

For an $n \times k$ matrix $\boldsymbol{M}$, let $\boldsymbol{M}_{i}$ denote the $i$ th row of $\boldsymbol{M}$, and $\boldsymbol{M}[j]$ its $j$ th
column. For a set $E \subseteq[n]$, let $\boldsymbol{M}_{E}$ denote the $|E| \times k$ submatrix of $\boldsymbol{M}$ formed by rows of $\boldsymbol{M}$ which are indexed by the elements of $E$. For a set $F \subseteq[k]$, let $\boldsymbol{M}[F]$ denote the $n \times|F|$ submatrix of $\boldsymbol{M}$ formed by columns of $\boldsymbol{M}$ which are indexed by the elements of $F$. For a matrix $\boldsymbol{M}$ over $\mathbb{F}_{q}$, we use $\operatorname{rank}_{q}(\boldsymbol{M})$ to denote the rank of $\boldsymbol{M}$ over $\mathbb{F}_{q}$.

For a set of vectors $\mathcal{S}$ over $\mathbb{F}_{q}$, we use notation $\operatorname{span}_{q}(\mathcal{S})$ to denote the linear space spanned over $\mathbb{F}_{q}$ by the vectors in $\mathcal{S}$.

### 1.4.3 Graph Theory Terminology

A simple graph is a pair $\mathcal{G}=(\mathcal{V}(\mathcal{G}), \mathcal{E}(\mathcal{G}))$ where $\mathcal{V}(\mathcal{G})$ is the set of vertices of $\mathcal{G}$ and $\mathcal{E}(\mathcal{G})$ is a set of unordered pairs of distinct vertices of $\mathcal{G}$. We refer to $\mathcal{E}(\mathcal{G})$ as the set of edges of $\mathcal{G}$. A typical edge of $\mathcal{G}$ is of the form $\{u, v\}$ where $u \in \mathcal{V}(\mathcal{G}), v \in \mathcal{V}(\mathcal{G})$, and $u \neq v$. If $e=\{u, v\} \in \mathcal{E}(\mathcal{G})$ we say that $u$ and $v$ are adjacent. We also refer to $u$ and $v$ as the endpoints of $e$.

A simple digraph is a pair $\mathcal{D}=(\mathcal{V}(\mathcal{D}), \mathcal{E}(\mathcal{D}))$ where $\mathcal{V}(\mathcal{D})$ is the set of vertices of $\mathcal{D}$, and $\mathcal{E}(\mathcal{D})$ is a set of ordered pairs of distinct vertices of $\mathcal{D}$. We refer to $\mathcal{E}(\mathcal{D})$ as the set of arcs (or directed edges) of $\mathcal{D}$. A typical arc of $\mathcal{D}$ is of the form $e=(u, v)$ where $u \in \mathcal{V}(\mathcal{D}), v \in \mathcal{V}(\mathcal{D})$, and $u \neq v$. The vertices $u$ and $v$ are called the endpoints of the arc $e$. The arc $e$ is called an out-going arc of $u$ and an in-coming arc of $v$.

Simple (di)graphs have no loops and no parallel (arcs) edges. In this thesis, only simple (di)graphs are considered. Therefore, we use (di)graphs to refer to simple (di)graphs for succinctness.

For a digraph $\mathcal{D}=(\mathcal{V}(\mathcal{D}), \mathcal{E}(\mathcal{D}))$, unless specified otherwise, we label the vertices of $\mathcal{D}$ by the natural numbers $1,2, \ldots,|\mathcal{V}(\mathcal{D})|$. In other words, if $|\mathcal{V}(\mathcal{D})|=n$ then we take $\mathcal{V}(\mathcal{D})=[n]$. The number of vertices $|\mathcal{V}(\mathcal{D})|$ is called the order of $\mathcal{D}$, whereas the number of $\operatorname{arcs}|\mathcal{E}(\mathcal{D})|$ is called the size of $\mathcal{D}$. The complement of a digraph
$\mathcal{D}=(\mathcal{V}(\mathcal{D}), \mathcal{E}(\mathcal{D}))$, denoted by $\overline{\mathcal{D}}=(\mathcal{V}(\overline{\mathcal{D}}), \mathcal{E}(\overline{\mathcal{D}}))$, is defined as follows. The vertex set is $\mathcal{V}(\overline{\mathcal{D}})=\mathcal{V}(\mathcal{D})$. The arc set is

$$
\mathcal{E}(\overline{\mathcal{D}})=\{(u, v): u \in \mathcal{V}(\mathcal{D}), v \in \mathcal{V}(\mathcal{D}), u \neq v,(u, v) \notin \mathcal{E}(\mathcal{D})\}
$$

Analogous conventions and concepts are also defined for graphs.
Given a digraph $\mathcal{D}=(\mathcal{V}(\mathcal{D}), \mathcal{E}(\mathcal{D}))$, the graph $\mathcal{G}=(\mathcal{V}(\mathcal{G}), \mathcal{E}(\mathcal{G}))$ where $\mathcal{V}(\mathcal{G})=$ $\mathcal{V}(\mathcal{D})$ and $\mathcal{E}(\mathcal{G})=\{\{u, v\}:(u, v) \in \mathcal{E}(\mathcal{D})\}$ is called the underlying graph of $\mathcal{D}$. Given a graph $\mathcal{G}$, the digraph $\mathcal{D}^{\mathcal{G}}$ obtained from $\mathcal{G}$ by replacing each edge $\{u, v\}$ of $\mathcal{G}$ by two $\operatorname{arcs}(u, v)$ and $(v, u)$ is called the digraph corresponding to $\mathcal{G}$.

A digraph $\mathcal{D}=(\mathcal{V}(\mathcal{D}), \mathcal{E}(\mathcal{D}))$ is called symmetric if it satisfies the property that $(u, v) \in \mathcal{E}(\mathcal{D})$ if and only if $(v, u) \in \mathcal{E}(\mathcal{D})$. A complete graph is a graph that contains all possible edges. A complete digraph is a digraph that contains all possible arcs. In other words, it is a symmetric digraph whose underlying graph is a complete graph.

A graph $\mathcal{G}=(\mathcal{V}(\mathcal{G}), \mathcal{E}(\mathcal{G}))$ is called bipartite if $\mathcal{V}(\mathcal{G})$ can be partitioned into two subsets $A$ and $B$ such that for every edge $\{u, v\} \in \mathcal{E}(\mathcal{G})$, it holds that $u \in A$ and $v \in B$, or vice versa.

A subgraph of a graph $\mathcal{G}$ (digraph $\mathcal{D}$, respectively) is a graph (digraph, respectively) whose vertex set $V$ is a subset of that of $\mathcal{G}(\mathcal{D}$, respectively) and whose edge set (arc set, respectively) is a subset of that of $\mathcal{G}(\mathcal{D}$, respectively) restricted on the vertices in $V$.

Let $V$ be a subset of vertices of a graph $\mathcal{G}=(\mathcal{V}(\mathcal{G}), \mathcal{E}(\mathcal{G}))$ (digraph $\mathcal{D}=$ $(\mathcal{V}(\mathcal{D}), \mathcal{E}(\mathcal{D}))$, respectively). The subgraph of $\mathcal{G}(\mathcal{D}$, respectively) induced by $V$ is a graph (digraph, respectively) whose vertex set is $V$, and edge set (arc set, respectively) is $\{\{u, v\}: u \in V, v \in V,\{u, v\} \in \mathcal{E}(\mathcal{G})\}(\{(u, v): u \in V, v \in$ $V,(u, v) \in \mathcal{E}(\mathcal{D})\})$. We refer to such a subgraph as an induced subgraph of $\mathcal{G}(\mathcal{D}$,
respectively).
A path in a graph $\mathcal{G}$ (digraph $\mathcal{D}$, respectively) is a sequence of distinct vertices $\left(u_{1}, u_{2}, \ldots, u_{k}\right)$, such that $\left\{u_{i}, u_{i+1}\right\} \in \mathcal{E}(\mathcal{G})\left(\left(u_{i}, u_{i+1}\right) \in \mathcal{E}(\mathcal{D})\right.$, respectively) for all $i \in[k-1]$.

A circuit (cycle, respectively) in a digraph $\mathcal{D}=(\mathcal{V}(\mathcal{D}), \mathcal{E}(\mathcal{D}))$ (graph $\mathcal{G}=$ $(\mathcal{V}(\mathcal{G}), \mathcal{E}(\mathcal{G}))$, respectively) is a sequence of pairwise distinct vertices

$$
\mathcal{C}=\left(i_{1}, i_{2}, \ldots, i_{\ell}\right),
$$

where $\left(i_{s}, i_{s+1}\right) \in \mathcal{E}(\mathcal{D})\left(\left\{i_{s}, i_{s+1}\right\} \in \mathcal{E}(\mathcal{G})\right.$, respectively) for all $s \in[\ell-1]$ and $\left(i_{\ell}, i_{1}\right) \in \mathcal{E}(\mathcal{D})\left(\left\{i_{\ell}, i_{1}\right\} \in \mathcal{E}(\mathcal{G})\right.$, respectively) as well. Let $\mathcal{V}(\mathcal{C})=\left\{i_{1}, i_{2}, \ldots, i_{\ell}\right\}$ be the set of vertices of $\mathcal{C}$. A digraph (graph, respectively) is called acyclic if it contains no circuits (cycles, respectively).

A (di)graph is called (strongly) connected if there is a path from each vertex in the (di)graph to every other vertex. The (strongly) connected components of a (di)graph are its maximal (strongly) connected subgraphs.

An edge $e$ of a graph $\mathcal{G}$ is a bridge of $\mathcal{G}$ if $\mathcal{G}-e$, the graph obtained from $\mathcal{G}$ by removing $e$, has more connected components than $\mathcal{G}$. In particular, an edge $e$ in a connected graph $\mathcal{G}$ is a bridge if and only if its removal renders the graph disconnected. A connected graph without any bridge (bridgeless) is called 2-edge connected.

A cut $C=(S, T)$ of a graph $\mathcal{G}$ is a partition of the vertex set $\mathcal{V}(\mathcal{G})$ of $\mathcal{G}$ into two parts $S$ and $T$. The cut-set of a cut $C=(S, T)$ is the set

$$
\{\{u, v\} \in \mathcal{E}(\mathcal{G}): u \in S, v \in T\} .
$$

The size of a cut is the number of edges in its cut-set. A min-cut is a cut whose size is not larger than the size of any other cut. If a connected graph is bridgeless then the size of each min-cut is at least two.

If $(u, v)$ is an arc in a digraph $\mathcal{D}=(\mathcal{V}(\mathcal{D}), \mathcal{E}(\mathcal{D}))$, then $v$ is called an out-neighbor of $u$. The set of out-neighbors of a vertex $u$ in the digraph $\mathcal{D}$ is denoted by $N_{O}^{\mathcal{D}}(u)$. We simply use $N_{O}(u)$ whenever there is no potential confusion. We refer to $\left|N_{O}^{\mathcal{D}}(u)\right|$ as the out-degree of $u$ in $\mathcal{D}$, denoted by $\operatorname{deg}_{O}^{\mathcal{D}}(u)$. For a graph $\mathcal{G}$, we denote by $N^{\mathcal{G}}(u)$ the set of neighbors of $u$, namely, the set of vertices adjacent to $u$.

An independent set in a graph $\mathcal{G}$ is a set of vertices of $\mathcal{G}$ with no edges connecting any two of them. An independent set in $\mathcal{G}$ of largest cardinality is called a maximum independent set in $\mathcal{G}$. The cardinality of such a maximum independent set is referred to as the independence number of $\mathcal{G}$, denoted by $\alpha(\mathcal{G})$.

A clique of a (di)graph is a set of vertices that induces a complete subgraph of that (di)graph. A clique cover of a (di)graph is a set of cliques that partition its vertex set. A minimum clique cover of a (di)graph is a clique cover of minimum number of cliques. The number of cliques in such a minimum clique cover of a (di)graph is called the clique cover number of that (di)graph. We denote by $\mathrm{cc}(\mathcal{G})$ the clique cover number of a graph $\mathcal{G}$ and $\operatorname{cc}(\mathcal{D})$ the clique cover number of a digraph $\mathcal{D}$.

A tree is a connected acyclic graph. A rooted tree is a tree with one special vertex designated to be the root. In a rooted tree, the parent of a vertex $v$ is the vertex connected to it on the path from $v$ to the root. Every vertex except the root has a unique parent. If $v$ is the parent of a vertex $u$ then $u$ is the child of $v$.

Definition 1.4.2 ([34]). Let $\mathcal{D}=(\mathcal{V}(\mathcal{D}), \mathcal{E}(\mathcal{D}))$ be a digraph of order $n$.

1. An $n \times n$ matrix $\boldsymbol{M}=\left(m_{i, j}\right)$ over an arbitrary field $\mathbb{F}$ is said to fit $\mathcal{D}$ if

$$
\left\{\begin{array}{l}
m_{i, j} \neq 0, \quad i=j, \\
m_{i, j}=0, \quad i \neq j, \quad(i, j) \notin \mathcal{E}(\mathcal{D}) .
\end{array}\right.
$$

2. The minrank of $\mathcal{D}$ over $\mathbb{F}$ is defined to be

$$
\operatorname{minrk}_{\mathbb{F}}(\mathcal{D})=\min \left\{\operatorname{rank}_{\mathbb{F}}(\boldsymbol{M}): \boldsymbol{M} \text { is over } \mathbb{F} \text { and } \boldsymbol{M} \text { fits } \mathcal{D}\right\}
$$

When $\mathbb{F}=\mathbb{F}_{q}$, we simply write $\operatorname{minrk}_{q}(\mathcal{D})$ instead of $\operatorname{minrk}_{\mathbb{F}_{q}}(\mathcal{D})$. We also have analogous definitions for a graph.

A (directed) hypergraph is a pair $(\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ is a set of vertices and $\mathcal{E}$ is a set of hyperacrs. A hyperarc $e$ itself is an ordered pair $(T, H)$, where $T$ and $H$ are both subsets of $\mathcal{V}$. They respectively represent the tail and the head of the hyperarc $e$. If $T=\{v\}$, i.e., $T$ consists of only one vertex $v$, then we simply write $(v, H)$ instead of $(\{v\}, H)$.

### 1.4.4 Information Theory Terminology

Let $X$ and $Y$ be discrete random variables taking values in the sets $\Sigma_{X}$ and $\Sigma_{Y}$, respectively. Let $\operatorname{Pr}(X=x)$ denote the probability that $X$ takes a particular value $x \in \Sigma_{X}$. The following notation is standard in Information Theory (see [20] for the background).

The (binary) entropy of $X$ is defined as

$$
\mathrm{H}(X)=-\sum_{x \in \Sigma_{X}} \operatorname{Pr}(X=x) \times \log _{2} \operatorname{Pr}(X=x)
$$

The conditional entropy of $X$ given $Y$ is defined as

$$
\mathrm{H}(X \mid Y)=-\sum_{x \in \Sigma_{X}, y \in \Sigma_{Y}} \operatorname{Pr}(X=x, Y=y) \times \log _{2} \operatorname{Pr}(X=x \mid Y=y)
$$

This definition can be naturally extended to $\mathrm{H}\left(X \mid Y_{1}, Y_{2}, \ldots, Y_{n}\right)$ for $n$ discrete random variables $Y_{i}, i \in[n]$.

If the probability distribution of $X$ is unchanged given the knowledge of $Y$, that is, $\operatorname{Pr}(X=x \mid Y=y)=\operatorname{Pr}(X=x)$ for all $x \in \Sigma_{X}$ and $y \in \Sigma_{Y}$, then $\mathrm{H}(X \mid Y)=\mathrm{H}(X)$. Indeed,

$$
\begin{aligned}
\mathrm{H}(X \mid Y) & =-\sum_{x \in \Sigma_{X}}\left(\sum_{y \in \Sigma_{Y}} \operatorname{Pr}(X=x, Y=y)\right) \times \log _{2} \operatorname{Pr}(X=x) \\
& =-\sum_{x \in \Sigma_{X}} \operatorname{Pr}(X=x) \times \log _{2} \operatorname{Pr}(X=x) \\
& =\mathrm{H}(X) .
\end{aligned}
$$

## 2. SECURE INDEX CODE WITH SIDE INFORMATION

Part of the work in this chapter was presented in the 2011 IEEE Symposium on Information Theory [22]. We investigate the security aspects of the Index Coding with Side Information problem. Building on the results of Bar-Yossef et al. [3,4], the properties of linear index codes are further explored. The notion of weak security, considered by Bhattad and Narayanan [9] in the context of network coding, is generalized to block security. It is shown that the linear index code based on a matrix $\boldsymbol{L}$, whose column space $\mathcal{C}(\boldsymbol{L})$ has length $n$, minimum distance $d$ and dual distance $d^{\perp}$, is ( $d-1-t$ )-block secure (and hence also weakly secure) if the adversary knows in advance $t \leq d-2$ messages, and is completely insecure if the adversary knows in advance more than $n-d^{\perp}$ messages. Strong security is examined under the circumstance that the adversary possesses some $t$ messages in advance and eavesdrops at most $\mu$ transmissions. We prove that for sufficiently large $q$, an optimal linear index code that is strongly secure against such an adversary has length $\kappa_{q}+\mu$. Here $\kappa_{q}$ is a generalization (to the case of hypergraph) of the minrank over $\mathbb{F}_{q}$ of the side information digraph for the ICSI problem in its original formulation [3]. As mentioned earlier in Section 1.2, both block security and strong security are, in fact, special cases of a more general security model called security against guessing, which was also introduced in the work of Bhattad and Narayanan [9].

### 2.1 Index Coding and Some Basic Results

In Chapter 2 and Chapter 3, we follow the most general model studied by Alon et al. [2]. The Index Coding with Side Information (ICSI) problem considers the following communications scenario. There is a unique sender $S$, who has a vector of messages $\boldsymbol{x}=\left(x_{1}, x_{2}, \ldots, x_{n}\right) \in \mathbb{F}_{q}^{n}$ in his possession, which is a realized value of a random vector $\boldsymbol{X}=\left(X_{1}, X_{2}, \ldots, X_{n}\right)$. The random variables $X_{1}, X_{2}, \ldots, X_{n}$ hereafter are assumed to be independent uniformly distributed random variables over $\mathbb{F}_{q}$. There are also $m$ receivers $R_{1}, R_{2}, \ldots, R_{m}$. For each $i \in[m]$, the receiver $R_{i}$ has some side information, that is, $R_{i}$ has a subset of messages $\left\{x_{j}\right\}_{j \in \mathcal{X}_{i}}$, where $\mathcal{X}_{i} \subsetneq[n]$. In addition, each $R_{i}$, for $i \in[m]$, is interested in receiving the message $x_{f(i)}$, for some demand function $f:[m] \rightarrow[n]$. Here we assume that $f(i) \notin \mathcal{X}_{i}$ for all $i \in[m]$. The assumption that each receiver is interested in exactly one message is not a limitation of the model. In fact, we can consider an equivalent problem by splitting each receiver who requests multiple messages into multiple receivers, each of whom requests exactly one message and have the same set of side information [3,10]. Let $\mathcal{X}=\left(\mathcal{X}_{1}, \mathcal{X}_{2}, \ldots, \mathcal{X}_{m}\right)$. An instance of the ICSI problem (or an ICSI instance, for short) is given by a quadruple ( $m, n, \mathcal{X}, f$ ). It can also be conveniently described by a (directed) hypergraph [2].

Definition 2.1.1. Let ( $m, n, \mathcal{X}, f$ ) be an ICSI instance. The corresponding side information (directed) hypergraph $\mathcal{H}=\mathcal{H}(m, n, \mathcal{X}, f)$ is defined by the vertex set $\mathcal{V}(\mathcal{H})=[n]$ and the hyperarc set $\mathcal{E}(\mathcal{H})$, where

$$
\mathcal{E}(\mathcal{H})=\left\{\left(f(i), \mathcal{X}_{i}\right): i \in[m]\right\} .
$$

We often refer to ( $m, n, \mathcal{X}, f$ ) as an ICSI instance described by the hypergraph $\mathcal{H}$.

(a) $\mathcal{H}_{1}$

(b) $\mathcal{H}_{2}$

(c) $\mathcal{D}_{2}$

Fig. 2.1: The hypergraphs $\mathcal{H}_{1}, \mathcal{H}_{2}$ and the digraph $\mathcal{D}_{2}$

For instance, consider an ICSI instance where $n=3$ (three messages), $m=4$ (four receivers), $f(1)=1, f(2)=2, f(3)=3, f(4)=2, \mathcal{X}_{1}=\{2,3\}, \mathcal{X}_{2}=\{1\}$, $\mathcal{X}_{3}=\{1,2\}$, and $\mathcal{X}_{4}=\{3\}$. The hypergraph $\mathcal{H}_{1}$ that describes this instance has three vertices $1,2,3$, and has four hyperarcs. These are $e_{1}=(1,\{2,3\}), e_{2}=(2,1)$, $e_{3}=(3,\{1,2\})$, and $e_{4}=(2,3)$. This hypergraph is depicted in Fig. 2.1a.

Remark 2.1.2. In the original setting of the ICSI problem [3], we have $m=n$ and $f(i)=i$ for all $i \in[n]$. In that case, the corresponding side information hypergraph has precisely $n$ hyperarcs where each of them has a different origin vertex. Then it is simpler to describe such an ICSI instance by a digraph $\mathcal{D}=(\mathcal{V}(\mathcal{D})=[n], \mathcal{E}(\mathcal{D}))$, so-called side information digraph $[3,46]$. For each hyperarc $\left(i, \mathcal{X}_{i}\right)$ of $\mathcal{H}$, there will be $\left|\mathcal{X}_{i}\right| \operatorname{arcs}(i, j)$ of $\mathcal{D}$, for $j \in \mathcal{X}_{i}$. Equivalently, $\mathcal{E}(\mathcal{D})=\left\{(i, j): i, j \in[n], j \in \mathcal{X}_{i}\right\}$. In this case, we refer to $\mathcal{D}$ as the underlying digraph of the hypergraph $\mathcal{H}$.

As an example, let $\mathcal{H}_{2}$ (Fig. 2.1b) be the hypergraph obtained from $\mathcal{H}_{1}$ by deleting the hyperarc $e_{4}$. The hypergraph $\mathcal{H}_{2}$ describes the ICSI instance obtained from the aforementioned instance by removing the last receiver. This new ICSI instance can also be described by the side information digraph $\mathcal{D}_{2}$ (Fig. 2.1c).

Furthermore, when it satisfies that $x_{j} \in \mathcal{X}_{i}$ if and only if $x_{i} \in \mathcal{X}_{j}$ for all $i \neq j$, the side information digraph $\mathcal{D}$ is symmetric, that is, $(i, j) \in \mathcal{E}(\mathcal{D})$ if and only if $(j, i) \in \mathcal{E}(\mathcal{D})$. In that case, we may consider the underlying graph $\mathcal{G}$ of $\mathcal{D}$ instead,
which is referred to as the underlying graph of the hypergraph $\mathcal{H}$. We also call $\mathcal{G}$ the side information graph of the corresponding ICSI instance.

Definition 2.1.3. An index code over $\mathbb{F}_{q}$ for an ICSI instance ( $m, n, \mathcal{X}, f$ ) (or just an $(m, n, \mathcal{X}, f)$-IC over $\left.\mathbb{F}_{q}\right)$, is an encoding function

$$
\mathfrak{E}: \mathbb{F}_{q}^{n} \rightarrow \mathbb{F}_{q}^{N}
$$

such that for each receiver $R_{i}, i \in[m]$, there exists a decoding function

$$
\mathfrak{D}_{i}: \mathbb{F}_{q}^{N} \times \mathbb{F}_{q}^{\left|\mathcal{X}_{i}\right|} \rightarrow \mathbb{F}_{q}
$$

satisfying

$$
\forall \boldsymbol{x} \in \mathbb{F}_{q}^{n}: \mathfrak{D}_{i}\left(\mathfrak{E}(\boldsymbol{x}), \boldsymbol{x}_{\mathcal{X}_{i}}\right)=x_{f(i)} .
$$

We also refer to such an $(m, n, \mathcal{X}, f)$-IC over $\mathbb{F}_{q}$ as an $\mathcal{H}$-IC over $\mathbb{F}_{q}$, where $\mathcal{H}=$ $\mathcal{H}(m, n, \mathcal{X}, f)$ is the corresponding side information hypergraph. The parameter $N$ is called the length of the index code. In the scheme corresponding to this code, $S$ broadcasts a vector $\mathfrak{E}(\boldsymbol{x})$ of length $N$ over $\mathbb{F}_{q}$.

Definition 2.1.4. An index code of minimum length is called optimal.

Definition 2.1.5. A linear index code is an index code for which the encoding function $\mathfrak{E}$ is a linear transformation over $\mathbb{F}_{q}$. Such a code can be described as

$$
\forall \boldsymbol{x} \in \mathbb{F}_{q}^{n}: \mathfrak{E}(\boldsymbol{x})=\boldsymbol{x} \boldsymbol{L},
$$

where $\boldsymbol{L}$ is an $n \times N$ matrix over $\mathbb{F}_{q}$. The matrix $\boldsymbol{L}$ is called the matrix corresponding
to the index code $\mathfrak{E}$. We also refer to $\mathfrak{E}$ as the index code based on $\boldsymbol{L}$. Notice that the length of $\mathfrak{E}$ is the number of columns of $\boldsymbol{L}$.

Let $E \subseteq[n]$ and $\boldsymbol{u} \in \mathbb{F}_{q}^{n}$. Recall that $\boldsymbol{u} \triangleleft E$ means $\operatorname{supp}(\boldsymbol{u}) \subseteq E$. If some receiver knows $x_{j}$ for all $j \in E$ and $\boldsymbol{u} \triangleleft E$, then this receiver can compute the value of $\boldsymbol{x} \boldsymbol{u}^{\mathrm{T}}$.

Hereafter, we assume that the sets $\mathcal{X}_{i}$ 's for $i \in[m]$ are known to $S$. Moreover, we also assume that the index code $\mathfrak{E}$ is known to each receiver $R_{i}$ for $i \in[m]$. In practice this can be achieved by a preliminary communication session, when the knowledge of the sets $\mathcal{X}_{i}$ 's for $i \in[m]$ and of the code $\mathfrak{E}$ are disseminated between the participants of the scheme.

Let

$$
\mathcal{C}(\boldsymbol{L})=\operatorname{span}_{q}\left(\left\{\boldsymbol{L}[j]^{\mathrm{T}}\right\}_{j \in[N]}\right),
$$

the subspace spanned by the transposed columns of $\boldsymbol{L}$. The following lemma was implicitly formulated by Bar-Yossef et al. [3, 4] for the case where $m=n, f(i)=i$ for all $i \in[m]$, and $q=2$. This lemma specifies a sufficient condition on $\mathcal{C}(\boldsymbol{L})$ so that $\boldsymbol{L}$ corresponds to a linear $(m, n, \mathcal{X}, f)$-IC over $\mathbb{F}_{q}$. We reproduce this lemma with its proof in its general form for the sake of completeness of the presentation.

Lemma 2.1.6. Let $\boldsymbol{L}$ be an $n \times N$ matrix over $\mathbb{F}_{q}$ and suppose $S$ broadcast $\boldsymbol{x} \boldsymbol{L}$. Then, for each $i \in[m]$, the receiver $R_{i}$ can reconstruct $x_{f(i)}$ if there exists a vector $\boldsymbol{u}^{(i)} \in \mathbb{F}_{q}^{n}$ satisfying

1. $\boldsymbol{u}^{(i)} \triangleleft \mathcal{X}_{i}$;
2. $\boldsymbol{u}^{(i)}+\boldsymbol{e}_{f(i)} \in \mathcal{C}(\boldsymbol{L})$.

Proof. Assume that $\boldsymbol{u}^{(i)} \triangleleft \mathcal{X}_{i}$ and $\boldsymbol{u}^{(i)}+\boldsymbol{e}_{f(i)} \in \mathcal{C}(\boldsymbol{L})$. Since $\boldsymbol{u}^{(i)}+\boldsymbol{e}_{f(i)} \in \mathcal{C}(\boldsymbol{L})$, there
exists $\boldsymbol{\beta} \in \mathbb{F}_{q}^{N}$ such that

$$
\boldsymbol{u}^{(i)}+\boldsymbol{e}_{f(i)}=\boldsymbol{\beta} \boldsymbol{L}^{\mathrm{T}}
$$

By taking the transpose and pre-multiplying by $\boldsymbol{x}$, we obtain

$$
\boldsymbol{x}\left(\boldsymbol{u}^{(i)}+\boldsymbol{e}_{f(i)}\right)^{\mathrm{T}}=(\boldsymbol{x} \boldsymbol{L}) \boldsymbol{\beta}^{\mathrm{T}}
$$

Therefore,

$$
x_{f(i)}=\boldsymbol{x} \boldsymbol{e}_{f(i)}^{\mathrm{T}}=(\boldsymbol{x} \boldsymbol{L}) \boldsymbol{\beta}^{\mathrm{T}}-\boldsymbol{x} \boldsymbol{u}^{(i)^{\mathrm{T}}}
$$

Observe that $R_{i}$ is able to find $\boldsymbol{u}^{(i)}$ and $\boldsymbol{\beta}$ from the knowledge of $\boldsymbol{L}$. Moreover, $R_{i}$ is also able to compute $\boldsymbol{x} \boldsymbol{u}^{(i)^{\mathrm{T}}}$ since $\boldsymbol{u}^{(i)} \triangleleft \mathcal{X}_{i}$, and receives $\boldsymbol{x} \boldsymbol{L}$ from $S$. Therefore, $R_{i}$ is able to determine $x_{f(i)}$.

Remark 2.1.7. Lemma 2.1.6 implies that $\boldsymbol{L}$ corresponds to a linear ( $m, n, \mathcal{X}, f$ )-IC over $\mathbb{F}_{q}$ if $\mathcal{C}(\boldsymbol{L}) \supseteq \operatorname{span}_{q}\left(\left\{\boldsymbol{u}^{(i)}+\boldsymbol{e}_{f(i)}\right\}_{i \in[m]}\right)$, for some $\boldsymbol{u}^{(i)} \triangleleft \mathcal{X}_{i}, i \in[m]$. We show later in Corollary 2.2.6 that this condition is also necessary. Finding such an $\boldsymbol{L}$ with minimum number of columns by a careful selection of $\boldsymbol{u}^{(i)}$ 's is a difficult task (in fact, the corresponding decision problem is NP-complete $[3,53]$ ), which, however, yields a linear coding scheme with minimum number of transmissions.

### 2.2 Block Secure Linear Index Codes

### 2.2.1 Block Security and Weak Security

In this section, we assume the presence of an adversary $A$ who can listen to all transmissions. Assume that $S$ employs a linear index code based on $\boldsymbol{L}$. The adversary is assumed to possess side information $\left\{x_{j}\right\}_{j \in \mathcal{X}_{A}}$, where $\mathcal{X}_{A} \subsetneq[n]$. For short, we say that $A$ knows or possesses $\boldsymbol{x}_{\mathcal{X}_{A}}$. The strength of the adversary $A$ is defined to be
$\left|\mathcal{X}_{A}\right|$. Denote $\widehat{\mathcal{X}}_{A} \triangleq[n] \backslash \mathcal{X}_{A}$. Note that by listening to $S$, the adversary also knows $\boldsymbol{s}=\boldsymbol{x} \boldsymbol{L}$. We define below several levels of security for linear index codes.

Definition 2.2.1. Suppose that the sender $S$ possesses a vector of messages $\boldsymbol{x} \in \mathbb{F}_{q}^{n}$, which is a realized value of a random vector $\boldsymbol{X}$. An adversary $A$ possesses $\boldsymbol{x}_{\mathcal{X}_{A}}$. Consider a linear $(m, n, \mathcal{X}, f)$-IC over $\mathbb{F}_{q}$ based on $\boldsymbol{L}$.

1. For $B \subseteq \widehat{\mathcal{X}}_{A}$, the adversary is said to have no information about $\boldsymbol{x}_{B}$ if

$$
\begin{equation*}
\mathrm{H}\left(\boldsymbol{X}_{B} \mid \boldsymbol{X} \boldsymbol{L}, \boldsymbol{X}_{\mathcal{X}_{A}}\right)=\mathrm{H}\left(\boldsymbol{X}_{B}\right) \tag{2.1}
\end{equation*}
$$

In other words, despite the partial knowledge on $\boldsymbol{x}$ that the adversary has (his side information and the transmissions he eavesdrops), the symbols $\boldsymbol{x}_{B}$ still look completely random to him.
2. The index code is said to be $b$-block secure against $\mathcal{X}_{A}$ if for every $b$-subset $B \subseteq \widehat{\mathcal{X}}_{A}$ (a $b$-subset is a subset of size $b$ ), the adversary has no information about $\boldsymbol{x}_{B}$.
3. The index code is said to be b-block secure against all adversaries of strength $t(0 \leq t \leq n-1)$ if it is $b$-block secure against $\mathcal{X}_{A}$ for every $\mathcal{X}_{A} \subset[n]$, where $\left|\mathcal{X}_{A}\right|=t$.
4. The index code is said to be weakly secure against $\mathcal{X}_{A}$ if it is 1-block secure against $\mathcal{X}_{A}$. In other words, after listening to all transmissions, the adversary has no information about each particular message that he does not possess in the first place.
5. The index code is said to be weakly secure against all adversaries of strength $t$ $(0 \leq t \leq n-1)$ if it is weakly secure against $\mathcal{X}_{A}$ for every $t$-subset $\mathcal{X}_{A}$ of $[n]$.
6. The index code is said to be completely insecure against $\mathcal{X}_{A}$ if an adversary, who possesses $\left\{x_{i}\right\}_{i \in \mathcal{X}_{A}}$, by listening to all transmissions, is able to determine $x_{i}$ for all $i \in \widehat{\mathcal{X}}_{A}$.
7. The index code is said to be completely insecure against any adversary of strength $t(0 \leq t \leq n-1)$ if an adversary, who possesses an arbitrary set of $t$ messages, is always able to reconstruct all of the other $n-t$ messages after listening to all transmissions.

Remark 2.2.2. Even when the index code is $b$-block secure $(b \geq 1)$ as defined above, the adversary is still able to obtain information about dependencies between various $x_{i}$ 's in $\widehat{\mathcal{X}}_{A}$ (but he gains no information about any group of $b$ particular messages). This definition of $b$-block security is a generalization of that of weak security $[9,58]$. Obviously, if an index code is $b$-block secure against $\mathcal{X}_{A}(b \geq 1)$ then it is also weakly secure against $\mathcal{X}_{A}$, but the converse is not always true.

### 2.2.2 Necessary and Sufficient Conditions for Block Security

In the sequel, we consider the sets $B \subseteq[n]$, where $B \neq \varnothing$, and $E \subseteq[n]$, where $E \neq \varnothing$. Moreover, we assume that the sets $\mathcal{X}_{A}, B$, and $E$ are disjoint, and that they form a partition of $[n]$, namely $\mathcal{X}_{A} \cup B \cup E=[n]$. Hence, $\widehat{\mathcal{X}}_{A}=[n] \backslash \mathcal{X}_{\mathcal{A}}=B \cup E$. Here, $\mathcal{X}_{A}$ corresponds to the set of messages that the adversary $A$ possesses, $B$ corresponds to the set of messages that $A$ is trying to gain information about, and $E$ corresponds to the set of remaining messages.

Lemma 2.2.3. Assume that for all $\boldsymbol{u} \triangleleft \mathcal{X}_{A}$ and for all $\alpha_{i} \in \mathbb{F}_{q}, i \in B\left(\right.$ not all $\alpha_{i}$ 's are zeros),

$$
\begin{equation*}
\boldsymbol{u}+\sum_{i \in B} \alpha_{i} \boldsymbol{e}_{i} \notin \mathcal{C}(\boldsymbol{L}) \tag{2.2}
\end{equation*}
$$

Then,

1. for all $i \in B$ :

$$
\begin{equation*}
\boldsymbol{L}_{i} \in \operatorname{span}_{q}\left(\left\{\boldsymbol{L}_{j}\right\}_{j \in E}\right) \tag{2.3}
\end{equation*}
$$

2. the system

$$
\begin{equation*}
\boldsymbol{y} \boldsymbol{L}_{E}=\boldsymbol{w} \boldsymbol{L}_{B} \tag{2.4}
\end{equation*}
$$

has at least one solution $\boldsymbol{y} \in \mathbb{F}_{q}^{|E|}$ for every choice of $\boldsymbol{w} \in \mathbb{F}_{q}^{|B|}$.
Proof.

1. If $\operatorname{rank}_{q}\left(\boldsymbol{L}_{E}\right)=N$ then the first claim follows immediately. Otherwise, assume that $\operatorname{rank}_{q}\left(\boldsymbol{L}_{E}\right)<N$. As the $N$ columns of $\boldsymbol{L}_{E}$ are linearly dependent, there exists $\boldsymbol{y} \in \mathbb{F}_{q}^{N} \backslash\{\mathbf{0}\}$ such that $\boldsymbol{y} \boldsymbol{L}_{E}^{\mathrm{T}}=\mathbf{0}$.
(a) If for all such $\boldsymbol{y}$ and for all $i \in B$ we have $\boldsymbol{y} \boldsymbol{L}_{i}^{\mathrm{T}}=0$, then

$$
\boldsymbol{L}_{i} \in\left(\left(\operatorname{span}_{q}\left(\left\{\boldsymbol{L}_{j}\right\}_{j \in E}\right)\right)^{\perp}\right)^{\perp}=\operatorname{span}_{q}\left(\left\{\boldsymbol{L}_{j}\right\}_{j \in E}\right)
$$

for all $i \in B$.
(b) Otherwise, there exist $\boldsymbol{y} \in \mathbb{F}_{q}^{N}$ and $i \in B$ such that $\boldsymbol{y} \boldsymbol{L}_{E}^{\mathrm{T}}=\mathbf{0}$ and $\boldsymbol{y} \boldsymbol{L}_{i}^{\mathrm{T}} \neq 0$. Without loss of generality, assume that

$$
\boldsymbol{L}=\left[\begin{array}{c}
\boldsymbol{L}_{\mathcal{X}_{A}} \\
\hline \boldsymbol{L}_{B} \\
\hline \boldsymbol{L}_{E}
\end{array}\right]
$$

Let $\boldsymbol{c}=\boldsymbol{y} \boldsymbol{L}^{\mathrm{T}} \in \mathcal{C}(\boldsymbol{L})$. Then

$$
\boldsymbol{c}=\left(\boldsymbol{c}_{\mathcal{X}_{A}}\left|\boldsymbol{c}_{B}\right| \boldsymbol{c}_{E}\right)=\left(\boldsymbol{y} \boldsymbol{L}_{\mathcal{X}_{A}}^{\mathrm{T}}\left|\boldsymbol{y} \boldsymbol{L}_{B}^{\mathrm{T}}\right| \boldsymbol{y} \boldsymbol{L}_{E}^{\mathrm{T}}\right)
$$

Hence $\boldsymbol{c}_{B}=\boldsymbol{y} \boldsymbol{L}_{B}^{\mathrm{T}} \neq \mathbf{0}$ and $\boldsymbol{c}_{E}=\boldsymbol{y} \boldsymbol{L}_{E}^{\mathrm{T}}=\mathbf{0}$. Let $\boldsymbol{u}=\left(\boldsymbol{c}_{\mathcal{X}_{A}}|\mathbf{0}| \mathbf{0}\right) \triangleleft \mathcal{X}_{A}$ and $\alpha_{i}=c_{i}$ for all $i \in B$. Then $\alpha_{i}$ 's are not all zero and $\boldsymbol{u}+\sum_{i \in B} \alpha_{i} \boldsymbol{e}_{i}=\boldsymbol{c} \in$ $\mathcal{C}(\boldsymbol{L})$, which contradicts (2.2).
2. By (2.3), each row of $\boldsymbol{L}_{B}$ is a linear combination of rows of $\boldsymbol{L}_{E}$. Hence $\boldsymbol{w} \boldsymbol{L}_{B}$ is also a linear combination of rows of $\boldsymbol{L}_{E}$. Therefore, (2.4) has at least one solution.

The following lemma provides us with a criterion to decide whether the index code based on a particular matrix $\boldsymbol{L}$ is block secure against an adversary $A$.

Lemma 2.2.4. Suppose that $S$ employs a linear $(m, n, \mathcal{X}, f)$-IC over $\mathbb{F}_{q}$ based on $\boldsymbol{L}$. For a subset $B \subseteq \widehat{\mathcal{X}}_{A}=[n] \backslash \mathcal{X}_{A}$, the adversary, after listening to all transmissions, has no information about $\boldsymbol{x}_{B}$ if and only if

$$
\begin{align*}
& \forall \boldsymbol{u} \triangleleft \mathcal{X}_{A}, \forall \alpha_{i} \in \mathbb{F}_{q} \text { with } \alpha_{i}, i \in B, \text { not all zero: } \\
& \qquad \boldsymbol{u}+\sum_{i \in B} \alpha_{i} \boldsymbol{e}_{i} \notin \mathcal{C}(\boldsymbol{L}) \tag{2.5}
\end{align*}
$$

In particular, for each $i \notin \mathcal{X}_{A}$, the adversary $A$ has no information about $x_{i}$ if and only if

$$
\forall \boldsymbol{u} \triangleleft \mathcal{X}_{A}: \boldsymbol{u}+\boldsymbol{e}_{i} \notin \mathcal{C}(\boldsymbol{L})
$$

Proof. Assume that (2.5) holds. We need to show that the entropy of $\boldsymbol{X}_{B}$ is not changed given the knowledge of $\boldsymbol{X} \boldsymbol{L}$ and $\boldsymbol{X}_{\mathcal{X}_{A}}$. It suffices to show that for all $\boldsymbol{g} \in \mathbb{F}_{q}^{|B|}:$

$$
\begin{equation*}
\operatorname{Pr}\left(\boldsymbol{X}_{B}=\boldsymbol{g} \mid \boldsymbol{X} \boldsymbol{L}=\boldsymbol{s}, \quad \boldsymbol{X}_{\mathcal{X}_{A}}=\boldsymbol{x}_{\mathcal{X}_{A}}\right)=\frac{1}{q^{|B|}} \tag{2.6}
\end{equation*}
$$

where $\boldsymbol{x} \in \mathbb{F}_{q}^{n}$ and $\boldsymbol{s}=\boldsymbol{x} \boldsymbol{L}$.

Consider the following linear system with the unknown $\boldsymbol{z} \in \mathbb{F}_{q}^{n}$

$$
\left\{\begin{array}{l}
z_{B}=\boldsymbol{g} \\
z_{\mathcal{X}_{A}}=\boldsymbol{x}_{\mathcal{X}_{A}} \\
z \boldsymbol{L}=s
\end{array}\right.
$$

which is equivalent to

$$
\left\{\begin{array}{l}
z_{B}=\boldsymbol{g}  \tag{2.7}\\
\boldsymbol{z}_{\mathcal{X}_{A}}=\boldsymbol{x}_{\mathcal{X}_{A}} \\
\boldsymbol{z}_{E} \boldsymbol{L}_{E}=s-\boldsymbol{g} \boldsymbol{L}_{B}-\boldsymbol{x}_{\mathcal{X}_{A}} \boldsymbol{L}_{\mathcal{X}_{A}}
\end{array}\right.
$$

In order to prove that (2.6) holds, it suffices to show that for all choices of $\boldsymbol{g} \in \mathbb{F}_{q}^{|B|},(2.7)$ always has the same number of solutions $\boldsymbol{z}$. Notice that the number of solutions $\boldsymbol{z}$ of (2.7) is equal to the number of solutions $\boldsymbol{z}_{E}$ of

$$
\begin{equation*}
\boldsymbol{z}_{E} \boldsymbol{L}_{E}=\boldsymbol{s}-\boldsymbol{g} \boldsymbol{L}_{B}-\boldsymbol{x}_{\mathcal{X}_{A}} \boldsymbol{L}_{\mathcal{X}_{A}} \tag{2.8}
\end{equation*}
$$

where $\boldsymbol{s}, \boldsymbol{g}$, and $\boldsymbol{x}_{\mathcal{X}_{\boldsymbol{A}}}$ are known. For any $\boldsymbol{g} \in \mathbb{F}_{q}^{|B|}$, if (2.8) has a solution, then it has exactly $q^{|E|-\text { rank }_{q}\left(\boldsymbol{L}_{E}\right)}$ different solutions. Therefore, it suffices to prove that (2.8) has at least one solution for every $\boldsymbol{g} \in \mathbb{F}_{q}^{|B|}$.

Since $\boldsymbol{s}=\boldsymbol{x} \boldsymbol{L}$, we have

$$
\begin{equation*}
\boldsymbol{x}_{E} \boldsymbol{L}_{E}=s-\boldsymbol{x}_{B} \boldsymbol{L}_{B}-\boldsymbol{x}_{\mathcal{X}_{A}} \boldsymbol{L}_{\mathcal{X}_{A}} . \tag{2.9}
\end{equation*}
$$

Subtracting (2.9) from (2.8) we obtain

$$
\left(\boldsymbol{z}_{E}-\boldsymbol{x}_{E}\right) \boldsymbol{L}_{E}=\left(\boldsymbol{x}_{B}-\boldsymbol{g}\right) \boldsymbol{L}_{B},
$$

which can be rewritten as

$$
\begin{equation*}
\boldsymbol{y} \boldsymbol{L}_{E}=\boldsymbol{w} \boldsymbol{L}_{B} \tag{2.10}
\end{equation*}
$$

where $\boldsymbol{y} \triangleq \boldsymbol{z}_{E}-\boldsymbol{x}_{E}$ and $\boldsymbol{w} \triangleq \boldsymbol{x}_{B}-\boldsymbol{g}$. Due to Lemma 2.2.3, (2.10) always has a solution $\boldsymbol{y}$, for every choice of $\boldsymbol{w}$. Therefore (2.8) has at least one solution for every $\boldsymbol{g} \in \mathbb{F}_{q}^{|B|}$.

Now we prove the converse. Assume that (2.5) does not hold. Then there exist $\boldsymbol{u} \triangleleft \mathcal{X}_{A}$ and $\alpha_{i} \in \mathbb{F}_{q}$ for $i \in B$, where $\alpha_{i}$ 's for $i \in B$ are not all zero, such that

$$
\sum_{i \in B} \alpha_{i} \boldsymbol{e}_{i}=\boldsymbol{c}-\boldsymbol{u}
$$

for some $\boldsymbol{c} \in \mathcal{C}(\boldsymbol{L})$. Hence, similarly to the proof of Lemma 2.1.6, the adversary obtains

$$
\sum_{i \in B} \alpha_{i} x_{i}=\boldsymbol{x}\left(\sum_{i \in B} \alpha_{i} \boldsymbol{e}_{i}\right)^{\mathrm{T}}=\boldsymbol{x}(\boldsymbol{c}-\boldsymbol{u})^{\mathrm{T}}=\boldsymbol{x} \boldsymbol{c}^{\mathrm{T}}-\boldsymbol{x} \boldsymbol{u}^{\mathrm{T}}
$$

Note that the adversary can calculate $\boldsymbol{x} \boldsymbol{c}^{\mathrm{T}}$ from $\boldsymbol{s}$, and can also find $\boldsymbol{x} \boldsymbol{u}^{\mathrm{T}}$ based on his own side information. Therefore, $A$ is able to compute a nontrivial linear combination of $x_{i}$ 's, $i \in B$. Hence the entropy $\mathrm{H}\left(\boldsymbol{X}_{B} \mid \boldsymbol{X} \boldsymbol{L}, \boldsymbol{X}_{\mathcal{X}_{A}}\right)<\mathrm{H}\left(\boldsymbol{X}_{B}\right)$. Thus, the adversary gains some information about the $\boldsymbol{x}_{B}$.

The following corollary generalizes Lemma 2.1 .6 by providing a necessary and sufficient condition for a receiver's ability to recover the desired message. Equivalently, it provides a necessary and sufficient condition for a receiver (or an adversary) to have no information about a particular message. Observe that, for one particular
message, a receiver (or an adversary) can recover that message or otherwise has no information about it.

Corollary 2.2.5. Let $\boldsymbol{L}$ be an $n \times N$ matrix over $\mathbb{F}_{q}$ and let $S$ broadcast $\boldsymbol{x} \boldsymbol{L}$. Then for each $i \in[m]$, the receiver $R_{i}$ can reconstruct $x_{f(i)}$ if and only if there exists a vector $\boldsymbol{u}^{(i)} \in \mathbb{F}_{q}^{n}$ such that

$$
\begin{aligned}
& \text { 1. } u^{(i)} \triangleleft \mathcal{X}_{i} \text {; } \\
& \text { 2. } u^{(i)}+\boldsymbol{e}_{f(i)} \in \mathcal{C}(\boldsymbol{L}) \text {. }
\end{aligned}
$$

Equivalently, the receiver $R_{i}$ (or any adversary that owns $\boldsymbol{x}_{\mathcal{X}_{i}}$ ) has no information about $x_{f(i)}$ if and only if the above condition is not satisfied.

Proof. The second statement is a reformulation of the last assertion in Lemma 2.2.4. The first statement is proved as follows. Suppose that there exists a vector $\boldsymbol{u}^{(i)} \in \mathbb{F}_{q}^{n}$ satisfying the two stated conditions. By Lemma 2.1.6, the receiver $R_{i}$ can determine $x_{f(i)}$. Conversely, suppose that $R_{i}$ can reconstruct $x_{f(i)}$. Then the aforementioned condition must be satisfied, for otherwise, $R_{i}$ would have no information about $x_{f(i)}$.

Corollary 2.2.6. The matrix $\boldsymbol{L}$ corresponds to a linear $(m, n, \mathcal{X}, f)$-IC over $\mathbb{F}_{q}$ if and only if for all $i \in[m]$, there exists a vector $\boldsymbol{u}^{(i)} \in \mathbb{F}_{q}^{n}$ satisfying

1. $\boldsymbol{u}^{(i)} \triangleleft \mathcal{X}_{i}$;
2. $\boldsymbol{u}^{(i)}+\boldsymbol{e}_{f(i)} \in \mathcal{C}(\boldsymbol{L})$.

Definition 2.2.7. Let $\mathcal{H}=\mathcal{H}(m, n, \mathcal{X}, f)$ be the hypergraph that describes an ICSI instance $(m, n, \mathcal{X}, f)$. The minrank over $\mathbb{F}_{q}$ of $\mathcal{H}$ is defined to be

$$
\begin{equation*}
\kappa_{q}(\mathcal{H})=\kappa_{q}(m, n, \mathcal{X}, f) \triangleq \min \left\{\operatorname{rank}_{q}\left(\left\{\boldsymbol{u}^{(i)}+\boldsymbol{e}_{f(i)}\right\}_{i \in[m]}\right): \boldsymbol{u}^{(i)} \in \mathbb{F}_{q}^{n}, \boldsymbol{u}^{(i)} \triangleleft \mathcal{X}_{i}\right\} \tag{2.11}
\end{equation*}
$$

We simply write $\kappa_{q}$ when there is no potential confusion.


Fig. 2.2: The hypergraph $\mathcal{H}_{1}$

For instance, let us consider the hypergraph $\mathcal{H}_{1}$ (Fig. 2.2) that describes the following ICSI instance: $n=3$ (three messages), $m=4$ (four receivers), $f(1)=1$, $f(2)=2, f(3)=3, f(4)=2, \mathcal{X}_{1}=\{2,3\}, \mathcal{X}_{2}=\{1\}, \mathcal{X}_{3}=\{1,2\}$, and $\mathcal{X}_{4}=\{3\}$. It is straightforward to verify that $\kappa_{2}\left(\mathcal{H}_{1}\right)=2$. We may select $\boldsymbol{u}^{(1)}=(0,1,1)$, $\boldsymbol{u}^{(2)}=(0,0,0), \boldsymbol{u}^{(3)}=(1,1,0)$, and $\boldsymbol{u}^{(4)}=(0,0,0)$. Then

$$
\begin{aligned}
\operatorname{rank}_{2}( & \left.\left\{\boldsymbol{u}^{(1)}+\boldsymbol{e}_{1}, \boldsymbol{u}^{(2)}+\boldsymbol{e}_{2}, \boldsymbol{u}^{(3)}+\boldsymbol{e}_{3}, \boldsymbol{u}^{(4)}+\boldsymbol{e}_{2}\right\}\right) \\
& \left.=\operatorname{rank}_{2}\{(1,1,1),(0,1,0),(1,1,1),(0,1,0))\right\} \\
& =2
\end{aligned}
$$

It follows from Corollary 2.2 .6 that $\boldsymbol{L}$ corresponds to a linear ( $m, n, \mathcal{X}, f$ )-IC over $\mathbb{F}_{q}$ if and only if $\mathcal{C}(\boldsymbol{L}) \supseteq \operatorname{span}_{q}\left(\left\{\boldsymbol{u}^{(i)}+\boldsymbol{e}_{f(i)}\right\}_{i \in[m]}\right)$, for some $\boldsymbol{u}^{(i)} \triangleleft \mathcal{X}_{i}, i \in[m]$. Therefore, $\kappa_{q}(m, n, \mathcal{X}, f)$ is the shortest possible length of a linear ( $m, n, \mathcal{X}, f$ )-IC over $\mathbb{F}_{q}$.

Corollary 2.2.8. The length of an optimal linear $(m, n, \mathcal{X}, f)$-IC over $\mathbb{F}_{q}$ is $\kappa_{q}=$ $\kappa_{q}(m, n, \mathcal{X}, f)$.

When $m=n$ and $f(i)=i$ for all $i \in[n]$, the quantity $\kappa_{q}$ is precisely the minrank over $\mathbb{F}_{q}$ of the side information digraph $\mathcal{D}$, which was introduced by Haemers [34] (see also [3]). Indeed, suppose that $\boldsymbol{u}^{(i)} \triangleleft \mathcal{X}_{i}$ for all $i \in[n]$ and let $\boldsymbol{A}=\left(a_{i, j}\right)$ be the $n \times n$ matrix whose $i$ th row is precisely $\boldsymbol{u}^{(i)}+\boldsymbol{e}_{i}, i \in[n]$. Recall from Remark 2.1.2 that $\mathcal{D}=(\mathcal{V}(\mathcal{D}), \mathcal{E}(\mathcal{D}))$, where $\mathcal{V}(\mathcal{D})=[n]$ and

$$
\mathcal{E}(\mathcal{D})=\left\{e=(i, j): i, j \in[n], j \in \mathcal{X}_{i}\right\} .
$$

Then $\boldsymbol{A}$ fits $\mathcal{D}$, that is,

$$
\left\{\begin{array}{l}
a_{i, j} \neq 0, \quad i=j \\
a_{i, j}=0, \quad i \neq j, \quad(i, j) \notin \mathcal{E}(\mathcal{D}) .
\end{array}\right.
$$

Conversely, if $\boldsymbol{A}^{\prime}$ fits $\mathcal{D}$ then by multiplying each row of $\boldsymbol{A}^{\prime}$ with a suitable nonzero constant (which does not change the rank of $\boldsymbol{A}^{\prime}$ ), one obtains a matrix $\boldsymbol{A}$ which is of the aforementioned form. Therefore, $\kappa_{q}$ defined as in (2.11) is indeed the minimum rank over $\mathbb{F}_{q}$ of a matrix that fits the side information graph $\mathcal{D}$. Thus, by Definition 1.4.2, $\kappa_{q}$ is precisely the minrank over $\mathbb{F}_{q}$ of $\mathcal{D}$.

Theorem 2.2.9. Consider a linear $(m, n, \mathcal{X}, f)$-IC over $\mathbb{F}_{q}$ based on L. Let d be the minimum distance of $\mathcal{C}(\boldsymbol{L})$.

1. This index code is $(d-1-t)$-block secure against all adversaries of strength $t \leq d-2$. In particular, it is weakly secure against all adversaries of strength $t=d-2$.
2. This index code is not weakly secure against at least one adversary of strength $t=d-1$. More generally, if there exists a codeword of $\mathcal{C}(\boldsymbol{L})$ of weight $w$, then this index code is not weakly secure against at least one adversary of strength $t=w-1$.
3. Every adversary of strength $t \leq d-1$ is able to find a list of $q^{n-t-N}$ vectors in $\mathbb{F}_{q}^{n}$ which includes the vector of messages $\boldsymbol{x}$.

Proof.

1. Assume that $t \leq d-2$. By Lemma 2.2.4, it suffices to show that for every $t$-subset $\mathcal{X}_{A}$ of $[n]$ and for every $(d-1-t)$-subset $B$ of $\widehat{\mathcal{X}}_{A}=[n] \backslash \mathcal{X}_{A}$,

$$
\begin{align*}
& \forall \boldsymbol{u} \triangleleft \mathcal{X}_{A}, \forall \alpha_{i} \in \mathbb{F}_{q} \text { with } \alpha_{i}, i \in B, \text { not all zero : } \\
& \qquad \boldsymbol{u}+\sum_{i \in B} \alpha_{i} \boldsymbol{e}_{i} \notin \mathcal{C}(\boldsymbol{L}) . \tag{2.12}
\end{align*}
$$

For such $\boldsymbol{u}$ and $\alpha_{i}{ }^{\prime}$ s, we have $\mathrm{wt}\left(\boldsymbol{u}+\sum_{i \in B} \alpha_{i} \boldsymbol{e}_{i}\right) \leq \mathrm{wt}(\boldsymbol{u})+\mathrm{wt}\left(\sum_{i \in B} \alpha_{i} \boldsymbol{e}_{i}\right) \leq$ $t+(d-1-t)=d-1<d$. Moreover, as $\operatorname{supp}(\boldsymbol{u}) \cap B=\varnothing$ and $\alpha_{i}{ }^{\prime}$ s, $i \in B$, are not all zero, we deduce that $\boldsymbol{u}+\sum_{i \in B} \alpha_{i} \boldsymbol{e}_{i} \neq \mathbf{0}$. We conclude that $\boldsymbol{u}+\sum_{i \in B} \alpha_{i} \boldsymbol{e}_{i} \notin \mathcal{C}(\boldsymbol{L})$.
2. We now show that the index code is not weakly secure against at least one adversary of strength $t=d-1$. The more general statement can be proved in an analogous way.

Pick a codeword $\boldsymbol{c}=\left(c_{1}, c_{2}, \ldots, c_{n}\right) \in \mathcal{C}(\boldsymbol{L})$ such that $\mathrm{wt}(\boldsymbol{c})=d$ and let $\operatorname{supp}(\boldsymbol{c})=\left\{i_{1}, i_{2}, \ldots, i_{d}\right\}$. Take $\mathcal{X}_{A}=\left\{i_{1}, i_{2}, \ldots, i_{d-1}\right\}$, where $\left|\mathcal{X}_{A}\right|=d-1$. Let

$$
\boldsymbol{u}=\left(\boldsymbol{c} / c_{i_{d}}-\boldsymbol{e}_{i_{d}}\right)
$$

Then, $\boldsymbol{u} \triangleleft \mathcal{X}_{A}$ and $\boldsymbol{u}+\boldsymbol{e}_{i_{d}}=\boldsymbol{c} / c_{i_{d}} \in \mathcal{C}(\boldsymbol{L})$. By Lemma 2.1.6, $A$ is able to determine $x_{i_{d}}$. Hence the index code is not weakly secure against the adversary $A$, who knows $d-1$ messages $x_{i}$ 's in advance.
3. Consider the following linear system of equations with unknown $\boldsymbol{z} \in \mathbb{F}_{q}^{n}$

$$
\left\{\begin{array}{l}
z_{\mathcal{X}_{A}}=x_{\mathcal{X}_{A}} \\
z L=s
\end{array}\right.
$$

which is equivalent to

$$
\left\{\begin{array}{l}
z_{\mathcal{X}_{A}}=x_{\mathcal{X}_{A}}  \tag{2.13}\\
z_{\widehat{\mathcal{X}}_{A}} \boldsymbol{L}_{\widehat{\mathcal{X}}_{A}}=s-\boldsymbol{x}_{\mathcal{X}_{A}} \boldsymbol{L}_{\mathcal{X}_{A}}
\end{array} .\right.
$$

The adversary $A$ attempts to solve this system. Given that $\boldsymbol{s}$ and $\boldsymbol{x}_{\mathcal{X}_{A}}$ are known, the system (2.13) has $n-t$ unknowns and $N$ equations. Note that $t \leq d-1$, and thus by applying Theorem 1.4.1 to $\mathcal{C}(\boldsymbol{L})$ we have $n-t \geq$ $n-d+1 \geq N$. If $\operatorname{rank}_{q}\left(\boldsymbol{L}_{\widehat{\mathcal{X}}_{A}}\right)=N$ then (2.13) has exactly $q^{n-t-N}$ solutions, as required.

Next, we show that $\operatorname{rank}_{q}\left(\boldsymbol{L}_{\widehat{\mathcal{X}}_{A}}\right)=N$. Assume, by contrary, that the $N$ (transposed) columns of $\boldsymbol{L}_{\widehat{\mathcal{X}}_{A}}$, denoted by $\boldsymbol{c}^{(1)}, \boldsymbol{c}^{(2)}, \ldots, \boldsymbol{c}^{(N)}$, are linearly dependent. Then there exist $\beta_{i} \in \mathbb{F}_{q}$ for $i \in[N]$, not all zero, such that $\sum_{i=1}^{N} \beta_{i} \boldsymbol{c}^{(i)}=\mathbf{0}$. Let

$$
\boldsymbol{c}=\sum_{i=1}^{N} \beta_{i} \boldsymbol{L}[i]^{\mathrm{T}} \in \mathcal{C}(\boldsymbol{L}) \backslash\{\mathbf{0}\}
$$

(Recall that $\boldsymbol{L}[i]$ denotes the $i$ th column of $\boldsymbol{L})$. Then $\boldsymbol{c}_{\widehat{\mathcal{X}}_{A}}=\sum_{i=1}^{N} \beta_{i} \boldsymbol{c}^{(i)}=\mathbf{0}$ and hence $\mathrm{wt}(\boldsymbol{c})=\mathrm{wt}\left(\boldsymbol{c}_{\mathcal{X}_{A}}\right) \leq t \leq d-1$. This is a contradiction, which follows
from the assumption that the $N$ rows of $\boldsymbol{L}_{\widehat{\mathcal{X}}_{A}}$ are linearly dependent.

Example 2.2.10. Let $q=2$. Assume that $\mathcal{X}_{A}=\varnothing$ and that $\mathcal{X}_{i} \neq \varnothing$ for all $i \in[m]$. For each $i \in[m]$ choose some $j_{i} \in \mathcal{X}_{i}$. Let $L$ be the binary matrix whose (transposed) columns form a basis of the space $\mathcal{C}(\boldsymbol{L})=\operatorname{span}_{q}\left(\left\{\boldsymbol{e}_{j_{i}}+\boldsymbol{e}_{f(i)}\right\}_{i \in[m]}\right)$. Then $\mathrm{d}(\mathcal{C}(\boldsymbol{L}))=2$. Since $t=\left|\mathcal{X}_{A}\right|=0$, we have $d-1-t=1$. Therefore by Theorem 2.2.9 the index code based on $\boldsymbol{L}$ is weakly secure against $A$. Moreover, by the Singleton bound, $\boldsymbol{L}$ has at most $N \leq n-d+1=n-1$ columns. In other words, the index code based on $\boldsymbol{L}$ requires at most $n-1$ transmissions.

### 2.2.3 Complete Insecurity

Theorem 2.2.9 provides a threshold for the security level of a linear index code based on $\boldsymbol{L}$. If $A$ has a prior knowledge of any $t \leq d-2$ messages, where $d=\mathrm{d}(\mathcal{C}(\boldsymbol{L}))$, then the index code is still secure, that is, the adversary has no information about any $d-1-t$ particular messages from $\left\{x_{j}\right\}_{j \in \widehat{\mathcal{X}}_{A}}$. On the other hand, the index code may no longer be secure against an adversary of strength $t=d-1$. The last assertion of Theorem 2.2.9 shows us the difference between being block secure and being strongly secure (we define the strong security rigorously later in Definition 2.4.2). More specifically, if the index code is strongly secure, the messages $\boldsymbol{x}_{\widehat{\mathcal{X}}_{A}}$, which are not leaked to the adversary in advance, look completely random to the adversary, that is, the probability to guess them correctly is $1 / q^{n-t}$. However, if the index code is $(d-1-t)$-block secure (for $t \leq d-2$ ), then the adversary is able to guess these messages correctly with probability $1 / q^{n-t-N}$.

For an adversary of strength $t \geq d$, the security of the index code depends on the properties of $\boldsymbol{L}$, in particular, it depends on the weight distribution of $\mathcal{C}(\boldsymbol{L})$. From Theorem 2.2.9, if there exists $\boldsymbol{c} \in \mathcal{C}(\boldsymbol{L})$ with $\operatorname{wt}(\boldsymbol{c})=w$, then the index code
is not weakly secure against some adversary of strength $t=w-1$. In general, the index code might still be (b-block or weakly) secure against some adversaries of strength $t$ for $t \geq d$. While we cannot make a general conclusion on the security of the index code when the adversary's strength is larger than $d-1$, Lemma 2.2.4 is still a useful tool to evaluate the security in that situation. However, as shown in Theorem 2.2.12, if the size of $\mathcal{X}_{A}$ is sufficiently large, then $A$ is able to determine all the messages in $\left\{x_{j}\right\}_{j \in \widehat{\mathcal{X}}_{A}}$. We first recall the following well-known result in coding theory.

Theorem 2.2.11 ([37], p. 66). Let $\mathcal{C}$ be an $[n, k, d]_{q}$ code with dual distance $d^{\perp}$ and $\boldsymbol{M}$ denote the $q^{k} \times n$ matrix whose $q^{k}$ rows are codewords of $\mathcal{C}$. If $r \leq d^{\perp}-1$ then each r-tuple from $\mathbb{F}_{q}$ appears in an arbitrary set of $r$ columns of $\boldsymbol{M}$ exactly $q^{k-r}$ times.

Theorem 2.2.12. The linear index code based on $\boldsymbol{L}$ is completely insecure against any adversary of strength $t \geq n-d^{\perp}+1$, where $d^{\perp}$ denotes the dual distance of $\mathcal{C}(\boldsymbol{L})$.

Proof. Suppose the adversary knows a subset $\left\{x_{j}\right\}_{j \in \mathcal{X}_{A}}$, where $\mathcal{X}_{A} \subsetneq[n]$ and $\left|\mathcal{X}_{A}\right|=$ $t \geq n-d^{\perp}+1$. By Corollary 2.2.5, it suffices to show that for all $j \in \hat{\mathcal{X}}_{A}$, there exists $\boldsymbol{u} \in \mathbb{F}_{q}^{n}$ satisfying simultaneously $\boldsymbol{u} \triangleleft \mathcal{X}_{A}$ and $\boldsymbol{u}+\boldsymbol{e}_{j} \in \mathcal{C}(\boldsymbol{L})$.

Indeed, take any $j \in \widehat{\mathcal{X}}_{A}$, and let $\rho=n-t \leq d^{\perp}-1$. Consider the $\rho$ indices that are not in $\mathcal{X}_{A}$. By Theorem 2.2.11, there exists a codeword $\boldsymbol{c} \in \mathcal{C}(\boldsymbol{L})$ with

$$
c_{\ell}= \begin{cases}1 & \text { if } \ell=j \\ 0 & \text { if } \ell \notin \mathcal{X}_{A} \cup\{j\}\end{cases}
$$

Then $\operatorname{supp}(\boldsymbol{c}) \subseteq \mathcal{X}_{A} \cup\{j\}$. We define $\boldsymbol{u} \in \mathbb{F}_{q}^{n}$ such that $\boldsymbol{u} \triangleleft \mathcal{X}_{A}$ as follows. For
$\ell \in \mathcal{X}_{A}$, we set $u_{\ell}=c_{\ell}$, and for $\ell \notin \mathcal{X}_{A}$, we set $u_{\ell}=0$. It is immediately clear that $\boldsymbol{c}=\boldsymbol{u}+\boldsymbol{e}_{j}$. Therefore, by Corollary 2.2.5, the adversary can reconstruct $x_{j}$. We have shown that the index code is completely insecure against an arbitrary set $\mathcal{X}_{A}$ satisfying $\left|\mathcal{X}_{A}\right| \geq n-d^{\perp}+1$, hence completing the proof.

When $\mathcal{C}(\boldsymbol{L})$ is an MDS code, we have $n-d^{\perp}+1=d-1$, and hence the two bounds established in Theorems 2.2.9 and 2.2.12 are actually tight. In that case, the third statement in Theorem 2.2.9 implies Theorem 2.2.12 as follows. This statement asserts that an adversary of strength $t=d-1$ can find a list of $q^{n-N-d+1}$ vectors that includes the vector of messages $\boldsymbol{x}$. Since $\mathcal{C}(\boldsymbol{L})$ is an MDS code, we have $n-N-d+1=0$. Therefore, the list contains only one element, namely $\boldsymbol{x}$ itself. Thus the index code is completely insecure against any adversary of strength $d-1$. The following example further illustrates the results stated in these theorems.

Example 2.2.13. Let $n=7, m=7, q=2$, and $f(i)=i$ for all $i \in[m]$. Suppose that the receivers have in their possession sets of messages as appear in the third column of the table below. Suppose also, that the demands of all receivers are as in the second column of the table.

| Receiver | Demand | $\left\{x_{j}\right\}_{j \in \mathcal{X}_{i}}$ |
| :---: | :---: | :---: |
| $R_{1}$ | $x_{1}$ | $\left\{x_{6}, x_{7}\right\}$ |
| $R_{2}$ | $x_{2}$ | $\left\{x_{5}, x_{7}\right\}$ |
| $R_{3}$ | $x_{3}$ | $\left\{x_{5}, x_{6}\right\}$ |
| $R_{4}$ | $x_{4}$ | $\left\{x_{5}, x_{6}, x_{7}\right\}$ |
| $R_{5}$ | $x_{5}$ | $\left\{x_{1}, x_{2}, x_{6}\right\}$ |
| $R_{6}$ | $x_{6}$ | $\left\{x_{1}, x_{3}, x_{4}\right\}$ |
| $R_{7}$ | $x_{7}$ | $\left\{x_{2}, x_{3}, x_{6}\right\}$ |

For $i \in[7]$, let $\boldsymbol{u}^{(i)} \in \mathbb{F}_{2}^{7}$ such that $\operatorname{supp}\left(\boldsymbol{u}^{(i)}\right)=\mathcal{X}_{i}$. Assume that an index code based on $\boldsymbol{L}$ with $\mathcal{C}(\boldsymbol{L})=\operatorname{span}_{q}\left(\left\{\boldsymbol{u}^{(i)}+\boldsymbol{e}_{i}\right\}_{i \in[7]}\right)$ is used. For instance, we can take $\boldsymbol{L}$ to be the matrix whose set of columns is $\left\{\boldsymbol{L}[i] \triangleq\left(\boldsymbol{u}^{(i)}+\boldsymbol{e}_{i}\right)^{\mathrm{T}}\right\}_{i \in[4]}$. It is easy to see that $\mathcal{C}(\boldsymbol{L})$ is a $[7,4,3]_{2}$ Hamming code with $d=3$ and $d^{\perp}=4$.

Following the coding scheme, $S$ broadcasts the following four bits:

$$
\begin{aligned}
& s_{1}=\boldsymbol{x}\left(\boldsymbol{u}^{(1)}+\boldsymbol{e}_{1}\right)^{\mathrm{T}}, \\
& s_{2}=\boldsymbol{x}\left(\boldsymbol{u}^{(2)}+\boldsymbol{e}_{2}\right)^{\mathrm{T}}, \\
& s_{3}=\boldsymbol{x}\left(\boldsymbol{u}^{(3)}+\boldsymbol{e}_{3}\right)^{\mathrm{T}}, \\
& s_{4}=\boldsymbol{x}\left(\boldsymbol{u}^{(4)}+\boldsymbol{e}_{4}\right)^{\mathrm{T}} .
\end{aligned}
$$

Each $R_{i}$ for $i \in[7]$, can compute $\boldsymbol{x}\left(\boldsymbol{u}^{(i)}+\boldsymbol{e}_{i}\right)^{\mathrm{T}}$ by using a linear combination of $s_{1}, s_{2}, s_{3}, s_{4}$. Then, each $R_{i}$ can subtract $\boldsymbol{x}\left(\boldsymbol{u}^{(i)}\right)^{\mathrm{T}}$ (his side information) from $\boldsymbol{x}\left(\boldsymbol{u}^{(i)}+\boldsymbol{e}_{i}\right)^{\mathrm{T}}$ to retrieve $x_{i}=\boldsymbol{x} \boldsymbol{e}_{i}^{\mathrm{T}}$.

For example, consider $R_{5}$. Since

$$
\boldsymbol{x}\left(\boldsymbol{u}^{(5)}+\boldsymbol{e}_{5}\right)^{\mathrm{T}}=\boldsymbol{x}\left(\left(\boldsymbol{u}^{(1)}+\boldsymbol{e}_{1}\right)+\left(\boldsymbol{u}^{(2)}+\boldsymbol{e}_{2}\right)\right)^{\mathrm{T}}=s_{1}+s_{2},
$$

$R_{5}$ subtracts $x_{1}+x_{2}+x_{6}$ from $s_{1}+s_{2}$ to obtain

$$
\left(s_{1}+s_{2}\right)-\left(x_{1}+x_{2}+x_{6}\right)=\left(x_{1}+x_{2}+x_{5}+x_{6}\right)-\left(x_{1}+x_{2}+x_{6}\right)=x_{5} .
$$

If an adversary $A$ has knowledge of a single message $x_{i}$, then by Theorem 2.2.9, $A$ is not able to determine any other message $x_{\ell}$, for $\ell \neq i$. Indeed, $\mathrm{d}(\mathcal{C}(\boldsymbol{L}))=3$, while $t=1$, the code is weakly secure against all adversaries of strength $t=1$. If none of the messages are leaked, then the adversary has no information about any group of 2 messages. On the other hand, the code is completely insecure against any adversary of strength $t \geq 4$; in that case $A$ is able to determine the remaining
$7-t$ messages.

### 2.3 Index Coding with Side and Restricted Information

Results on the block security of a linear index code can be employed to study the linear coding schemes for an extension of the ICSI problem, so-called the Index Coding with Side and Restricted Information (ICSRI) problem. An instance ( $m, n, \mathcal{X}, \mathcal{Z}, f$ ) of the ICSRI problem consists of almost the same parameters as that of the ICSI problem. The only new parameter, which is $\mathcal{Z}=\left(\mathcal{Z}_{1}, \mathcal{Z}_{2}, \ldots, \mathcal{Z}_{m}\right)$, represents the sets of messages that the receivers are not allowed to obtain. The goal is that at the end of the communication round, for each $i \in[m]$, the receiver $R_{i}$ obtains the message $x_{f(i)}$ and has no information about $x_{j}$ for all $j \in \mathcal{Z}_{i}$. The notion of a linear $(m, n, \mathcal{X}, f)$-IC over $\mathbb{F}_{q}$ is naturally extended to that of a linear $(m, n, \mathcal{X}, \mathcal{Z}, f)$-IC over $\mathbb{F}_{q}$.

Let

$$
\mathcal{F}(m, n, \mathcal{X}, \mathcal{Z}, f) \triangleq \bigcup_{i=1}^{m}\left\{\boldsymbol{u}+\boldsymbol{e}_{j}: \boldsymbol{u} \triangleleft \mathcal{X}_{i}, j \in \mathcal{Z}_{i}\right\}
$$

The following proposition provides a necessary and sufficient condition for a linear index code to be also a solution to an instance of the ICSRI problem.

Proposition 2.3.1. The linear $(m, n, \mathcal{X}, f)$-IC over $\mathbb{F}_{q}$ based on $\boldsymbol{L}$ is also a linear $(m, n, \mathcal{X}, \mathcal{Z}, f)-I C$ if and only if $\mathcal{C}(\boldsymbol{L}) \cap \mathcal{F}(m, n, \mathcal{X}, \mathcal{Z}, f)=\varnothing$.

Proof. Let $S$ employ the $(m, n, \mathcal{X}, f)$-IC over $\mathbb{F}_{q}$ based on $\boldsymbol{L}$. Then clearly $R_{i}$ can recover $x_{f(i)}$ for all $i \in[m]$. Due to Lemma 2.2.4, for each $i \in[m]$ and $j \in \mathcal{Z}_{i}$, the receiver $R_{i}$ has no information about $x_{j}$ if and only if

$$
\forall \boldsymbol{u} \triangleleft \mathcal{X}_{i}: \boldsymbol{u}+\boldsymbol{e}_{j} \notin \mathcal{C}(\boldsymbol{L})
$$

Hence we complete the proof.

Example 2.3.2. Consider an instance ( $m, n, \mathcal{X}, \mathcal{Z}, f$ ) of the ICSRI problem where $m, n, \mathcal{X}$, and $f$ are defined as in Example 2.2.13. Moreover, $\mathcal{Z}=\left(\mathcal{Z}_{1}, \mathcal{Z}_{2}, \ldots, \mathcal{Z}_{7}\right)$, where $\mathcal{Z}_{1}=\{2,3,4,5\}, \mathcal{Z}_{2}=\{1,3,4,6\}, \mathcal{Z}_{3}=\{1,2,4,7\}$, and $\mathcal{Z}_{4}=\mathcal{Z}_{5}=\mathcal{Z}_{6}=$ $\mathcal{Z}_{7}=\varnothing$. Consider the index code based on $\boldsymbol{L}$ constructed in Example 2.2.13. It is straightforward to verify that $\mathcal{C}(\boldsymbol{L}) \cap \mathcal{F}(m, n, \mathcal{X}, \mathcal{Z}, f)=\varnothing$. Therefore, by Proposition 2.3.1, this index code also provides a solution to this instance of the ICSRI problem.

## Let

$$
\bar{\kappa}_{q}=\bar{\kappa}_{q}(m, n, \mathcal{X}, \mathcal{Z}, f) \triangleq \min \left\{\operatorname{rank}_{q}\left(\left\{\boldsymbol{u}^{(i)}+\boldsymbol{e}_{f(i)}\right\}_{i \in[m]}\right)\right\}
$$

where the minimum is taken over all choices of $\boldsymbol{u}^{(i)} \triangleleft \mathcal{X}_{i}$ that satisfy

$$
\begin{equation*}
\operatorname{span}_{q}\left(\left\{\boldsymbol{u}^{(i)}+\boldsymbol{e}_{f(i)}\right\}\right) \cap \mathcal{F}(m, n, \mathcal{X}, \mathcal{Z}, f)=\varnothing \tag{2.14}
\end{equation*}
$$

Let $\bar{\kappa}_{q}=+\infty$ if there are no choices of $\boldsymbol{u}^{(i)}$ 's that satisfy (2.14). Then the following proposition is immediate.

Proposition 2.3.3. If $\bar{\kappa}_{q}<+\infty$ then the length of an optimal linear ( $m, n, \mathcal{X}, \mathcal{Z}, f$ )IC over $\mathbb{F}_{q}$ is $\bar{\kappa}_{q}$. If $\bar{\kappa}_{q}=+\infty$ then there does not exist a linear $(m, n, \mathcal{X}, \mathcal{Z}, f)$-IC over $\mathbb{F}_{q}$.

### 2.4 Strongly Secure Index Code with Side Information

In this section, we consider a less powerful adversary, who owns some prior side information and eavesdrops at most $\mu$ transmissions.

### 2.4.1 A Lower Bound on the Length of a Strongly Secure Index Code

We start this section with a generalization of the definition of index codes to that of randomized index codes. Consider $\eta \in \mathbb{N}$ random variables $G_{1}, G_{2}, \ldots, G_{\eta}$, which are distributed independently and uniformly over $\mathbb{F}_{q}$. Let $\boldsymbol{G}=\left(G_{1}, G_{2}, \ldots, G_{\eta}\right)$ and let $\boldsymbol{g}=\left(g_{1}, g_{2}, \ldots, g_{\eta}\right)$ be a realization of $\boldsymbol{G}$.

Definition 2.4.1. An $\eta$-randomized $(m, n, \mathcal{X}, f)$-IC over $\mathbb{F}_{q}$ for an ICSI instance $(m, n, \mathcal{X}, f)$ is an encoding function

$$
\mathfrak{E}: \mathbb{F}_{q}^{n} \times \mathbb{F}_{q}^{\eta} \rightarrow \mathbb{F}_{q}^{N},
$$

such that for each receiver $R_{i}, i \in[m]$, there exists a decoding function

$$
\mathfrak{D}_{i}: \mathbb{F}_{q}^{N} \times \mathbb{F}_{q}^{\left|\mathcal{X}_{i}\right|} \rightarrow \mathbb{F}_{q}
$$

satisfying

$$
\forall \boldsymbol{x} \in \mathbb{F}_{q}^{n}: \mathfrak{D}_{i}\left(\mathfrak{E}(\boldsymbol{x}, \boldsymbol{g}), \boldsymbol{x}_{\mathcal{X}_{i}}\right)=x_{f(i)},
$$

for every $\boldsymbol{g} \in \mathbb{F}_{q}^{\eta}$, which is a realization of the random vector $\boldsymbol{G}$.

An $\eta$-randomized index code is linear over $\mathbb{F}_{q}$ if it has a linear encoding function $\mathfrak{E}$,

$$
\mathfrak{E}(\boldsymbol{x}, \boldsymbol{g})=(\boldsymbol{x} \mid \boldsymbol{g}) \boldsymbol{L}
$$

where $\boldsymbol{L}$ is an $(n+\eta) \times N$ matrix over $\mathbb{F}_{q}$. Observe that by simply treating $x_{1}, x_{2}, \ldots, x_{n}, g_{1}, g_{2}, \ldots, g_{\eta}$ as messages, the results from previous sections still apply to linear randomized index codes.

Definition 2.4.2. The linear $\eta$-randomized $(m, n, \mathcal{X}, f)$-IC over $\mathbb{F}_{q}$ based on $\boldsymbol{L}$ is
said to be ( $\mu, t$ )-strongly secure if the following holds. An adversary $A$ who possesses $t$ arbitrary messages $\boldsymbol{x}_{\mathcal{X}_{A}}$, for some $t$-subset $\mathcal{X}_{A}$ of $[n]$, and is able to listen to at most $\mu$ among $N$ transmissions, gains no information about other messages. Equivalently,

$$
\mathrm{H}\left(\boldsymbol{X}_{\widehat{\mathcal{X}}_{A}} \mid \boldsymbol{X}_{\mathcal{X}_{A}},(\boldsymbol{X} \mid \boldsymbol{G}) \boldsymbol{L}[W]\right)=\mathrm{H}\left(\boldsymbol{X}_{\widehat{\mathcal{X}}_{A}}\right)
$$

for any $W \subseteq[N]$ and $|W| \leq \mu$.

Remark 2.4.3. If $\mu=t=\eta=0$, a ( $\mu, t)$-strongly secure $\eta$-randomized ( $m, n, \mathcal{X}, f$ )IC over $\mathbb{F}_{q}$ becomes a normal $(m, n, \mathcal{X}, f)$-IC over $\mathbb{F}_{q}$.

We henceforth assume that each message is requested by at least one receiver, for otherwise, that "useless" message can be discarded without doing any harm to the coding scheme.

Lemma 2.4.4. If $\boldsymbol{L}$ corresponds to a $(\mu, t)$-strongly secure linear $\eta$-randomized $(m, n, \mathcal{X}, f)$-IC over $\mathbb{F}_{q}$, then $\eta \geq \mu$.

Proof. We prove this lemma by contradiction. Suppose that $\boldsymbol{L}$ corresponds to a $(\mu, t)$-strongly secure $\eta$-randomized $(m, n, \mathcal{X}, f)$-IC over $\mathbb{F}_{q}$ and that $\eta<\mu$.

For a subset $W$ of $[N]$ let $\mathcal{C}(\boldsymbol{L}[W])$ be the space spanned by (transposed) columns of $\boldsymbol{L}$ indexed by elements of $W$. Let $E=\{n+1, n+2, \ldots, n+\eta\}$. For all subsets $W$ of $[N]$ with $|W| \leq \mu$, it holds that

$$
\mathrm{H}\left(\boldsymbol{X}_{\widehat{\mathcal{X}}_{A}} \mid(\boldsymbol{X} \mid \boldsymbol{G}) \boldsymbol{L}[W], \boldsymbol{X}_{\mathcal{X}_{A}}\right)=\mathrm{H}\left(\boldsymbol{X}_{\widehat{\mathcal{X}}_{A}}\right),
$$

that is, an adversary who owns $\boldsymbol{x}_{\mathcal{X}_{A}}$ gains no information about $\boldsymbol{x}_{\widehat{\mathcal{X}}_{A}}$ after eavesdropping $|W|$ transmissions. Therefore, applying Lemma 2.2 .4 with $\mathcal{C}(\boldsymbol{L})$ being replaced
by $\mathcal{C}(\boldsymbol{L}[W])$ (equivalently, $\boldsymbol{L}$ being replaced by $\boldsymbol{L}[W]$ ), we conclude that $\mathcal{C}(\boldsymbol{L}[W])$ does not contain a vector $\boldsymbol{c}$ that satisfies $\boldsymbol{c}_{\widehat{\mathcal{X}}_{A}} \neq \mathbf{0}$ and $\boldsymbol{c}_{E}=\mathbf{0}$. We refer to this property of $\mathcal{C}(\boldsymbol{L}[W])$ as Property A.

Let $\boldsymbol{L}^{\prime}=\left(\boldsymbol{L}_{\widehat{\mathcal{X}}_{A} \cup E}\right)^{\mathrm{T}}$ be the matrix obtained from $\boldsymbol{L}$ by first deleting rows of $\boldsymbol{L}$ indexed by $\mathcal{X}_{A}$, and then taking its transpose. We claim that $\operatorname{rank}_{q}\left(\boldsymbol{L}^{\prime}\right) \leq \mu-1$. Indeed, take any $\mu$ rows of $\boldsymbol{L}^{\prime}$, say $\boldsymbol{L}_{j_{1}}^{\prime}, \ldots, \boldsymbol{L}_{j_{\mu}}^{\prime}$, we show that these $\mu$ rows are linearly dependent. Let $\boldsymbol{L}^{\prime \prime}$ be the submatrix of $\boldsymbol{L}^{\prime}$ formed by the last $\eta$ columns. Since $\eta<\mu$, the $\mu$ rows $\boldsymbol{L}_{j_{1}}^{\prime \prime}, \ldots, \boldsymbol{L}_{j_{\mu}}^{\prime \prime}$ are linearly dependent. Hence there exist $\alpha_{1}, \alpha_{2}, \ldots, \alpha_{\mu}$, not all zeros, such that

$$
\sum_{\ell=1}^{\mu} \alpha_{\ell} \boldsymbol{L}_{j_{\ell}}^{\prime \prime}=\mathbf{0}
$$

which implies

$$
\sum_{\ell=1}^{\mu} \alpha_{\ell} \boldsymbol{L}_{j_{\ell}}^{\prime}=\mathbf{0}
$$

due to Property A. The claim follows.
Now let $r=\operatorname{rank}_{q}\left(\boldsymbol{L}^{\prime}\right)<\mu$, and let

$$
\left\{\boldsymbol{L}_{j_{1}}^{\prime}, \boldsymbol{L}_{j_{2}}^{\prime}, \ldots, \boldsymbol{L}_{j_{r}}^{\prime}\right\}
$$

be a basis of the space spanned by rows of $\boldsymbol{L}^{\prime}$. Suppose that the receiver $R_{i}$ requests $x_{f(i)}$ where $f(i) \in \widehat{\mathcal{X}}_{A}$. By Corollary 2.2.5, $\mathcal{C}(\boldsymbol{L})$ contains a vector $\boldsymbol{c}=\boldsymbol{u}+\boldsymbol{e}_{f(i)}$ where $\boldsymbol{u} \triangleleft \mathcal{X}_{i}$. Therefore, $\boldsymbol{c}_{E}=\mathbf{0}$ and $\boldsymbol{c}_{\widehat{\mathcal{X}}_{A}} \neq \mathbf{0}$. On the other hand, there exists $\beta_{1}, \beta_{2}, \ldots, \beta_{r}$ such that

$$
\left(\boldsymbol{c}_{\widehat{\mathcal{X}}_{A}} \mid \boldsymbol{c}_{E}\right)=\sum_{\ell=1}^{r} \beta_{\ell} \boldsymbol{L}_{j_{\ell}}^{\prime} .
$$

Since $r<\mu$ and $\boldsymbol{c}_{E}=\mathbf{0}$, by Property A we have $\boldsymbol{c}_{\widehat{\mathcal{X}}_{A}}=\mathbf{0}$, which is a contradiction.

Remark 2.4.5. By Lemma 2.4.4, a randomized index code requires at least $\mu$ random symbols in order to be $(\mu, t)$-strongly secure. We show in the next subsection that using $\mu$ random symbols is also sufficient.

Lemma 2.4.6. Supose that $\boldsymbol{L}$ corresponds to a linear $\mu$-randomized ( $m, n, \mathcal{X}, f$ )-IC over $\mathbb{F}_{q}$. If this randomized index code is $(\mu, t)$-strongly secure, then for all $i \in[\mu]$, there exists a vector $\boldsymbol{v}^{(i)} \in \mathbb{F}_{q}^{n+\mu}$ satisfying

$$
\begin{aligned}
& \text { 1. } \boldsymbol{v}^{(i)} \triangleleft[n] \text {; } \\
& \text { 2. } \boldsymbol{v}^{(i)}+\boldsymbol{e}_{n+i} \in \mathcal{C}(\boldsymbol{L}) \text {. }
\end{aligned}
$$

Proof. Assume, by contradiction, that for some $i \in[\mu]$, we have $\boldsymbol{v}^{(i)}+\boldsymbol{e}_{n+i} \notin \mathcal{C}(\boldsymbol{L})$ for all $\boldsymbol{v}^{(i)} \triangleleft[n]$. By Corollary 2.2.5, a virtual receiver with side information $\left\{x_{j}\right\}_{j \in[n]}$ would have no information about $g_{i}$ after receiving all transmissions. In other words, we have

$$
\begin{equation*}
\mathrm{H}\left(G_{i} \mid(\boldsymbol{X} \mid \boldsymbol{G}) \boldsymbol{L}, \boldsymbol{X}\right)=\mathrm{H}\left(G_{i}\right) \tag{2.15}
\end{equation*}
$$

and, in particular, for a smaller set of side information,

$$
\begin{equation*}
\mathrm{H}\left(G_{i} \mid(\boldsymbol{X} \mid \boldsymbol{G}) \boldsymbol{L}, \boldsymbol{X}_{\mathcal{X}_{A}}\right)=\mathrm{H}\left(G_{i}\right) \tag{2.16}
\end{equation*}
$$

Since the randomized index code is $(\mu, t)$-strongly secure, for every $\mu$-subset $W$ of [ $N]$ and every $t$-subset $\mathcal{X}_{A}$ of $[n]$, we have

$$
\begin{equation*}
\mathrm{H}\left(\boldsymbol{X}_{\widehat{\mathcal{X}}_{A}} \mid(\boldsymbol{X} \mid \boldsymbol{G}) \boldsymbol{L}[W], \boldsymbol{X}_{\mathcal{X}_{A}}\right)=\mathrm{H}\left(\boldsymbol{X}_{\widehat{\mathcal{X}}_{A}}\right) . \tag{2.17}
\end{equation*}
$$

We claim that if the realized value of $G_{i}$ is known to the adversary, this randomized
index code is still $(\mu, t)$-strongly secure. In other words, we aim to show that

$$
\begin{equation*}
\mathrm{H}\left(\boldsymbol{X}_{\widehat{\mathcal{X}}_{A}} \mid(\boldsymbol{X} \mid \boldsymbol{G}) \boldsymbol{L}[W], \boldsymbol{X}_{\mathcal{X}_{A}}, G_{i}\right)=\mathrm{H}\left(\boldsymbol{X}_{\widehat{\mathcal{X}}_{A}}\right), \tag{2.18}
\end{equation*}
$$

for every $\mu$-subset $W$ of $[N]$ and every $t$-subset $\mathcal{X}_{A}$ of $[n]$. Indeed, the left-hand side of (2.18) is equal to

$$
\mathrm{H}\left(\boldsymbol{X}_{\widehat{\mathcal{X}}_{A}} \mid(\boldsymbol{X} \mid \boldsymbol{G}) \boldsymbol{L}[W], \boldsymbol{X}_{\mathcal{X}_{A}}\right)-\mathrm{I}\left(\boldsymbol{X}_{\widehat{\mathcal{X}}_{A}} ; G_{i} \mid(\boldsymbol{X} \mid \boldsymbol{G}) \boldsymbol{L}[W], \boldsymbol{X}_{\mathcal{X}_{A}}\right),
$$

which is

$$
\mathrm{H}\left(\boldsymbol{X}_{\widehat{\mathcal{X}}_{A}}\right)-\mathrm{I}\left(\boldsymbol{X}_{\widehat{\mathcal{X}}_{A}} ; G_{i} \mid(\boldsymbol{X} \mid \boldsymbol{G}) \boldsymbol{L}[W], \boldsymbol{X}_{\mathcal{X}_{A}}\right)
$$

due to (2.17). Hence, it suffices to show that

$$
\mathrm{I}\left(\boldsymbol{X}_{\hat{\mathcal{X}}_{A}} ; G_{i} \mid(\boldsymbol{X} \mid \boldsymbol{G}) \boldsymbol{L}[W], \boldsymbol{X}_{\mathcal{X}_{A}}\right)=0 .
$$

Using (2.15) we have

$$
\begin{aligned}
\mathrm{I}\left(\boldsymbol{X}_{\widehat{\mathcal{X}}_{A}} ; G_{i} \mid(\boldsymbol{X} \mid \boldsymbol{G}) \boldsymbol{L}[W], \boldsymbol{X}_{\mathcal{X}_{A}}\right)= & \mathrm{H}\left(G_{i} \mid(\boldsymbol{X} \mid \boldsymbol{G}) \boldsymbol{L}[W], \boldsymbol{X}_{\mathcal{X}_{A}}\right) \\
& -\mathrm{H}\left(G_{i} \mid(\boldsymbol{X} \mid \boldsymbol{G}) \boldsymbol{L}[W], \boldsymbol{X}_{\mathcal{X}_{A}}, \boldsymbol{X}_{\widehat{\mathcal{X}}_{A}}\right) \\
= & \mathrm{H}\left(G_{i} \mid(\boldsymbol{X} \mid \boldsymbol{G}) \boldsymbol{L}[W], \boldsymbol{X}_{\mathcal{X}_{A}}\right) \\
& -\mathrm{H}\left(G_{i} \mid(\boldsymbol{X} \mid \boldsymbol{G}) \boldsymbol{L}[W], \boldsymbol{X}\right) \\
= & \mathrm{H}\left(G_{i}\right)-\mathrm{H}\left(G_{i}\right) \\
= & 0
\end{aligned}
$$

where the third transition is due to (2.15) and (2.16). Thus, we have shown that the randomized index code is still $(\mu, t)$-strongly secure if the adversary knows the
realized value of $G_{i}$. Equivalently, discarding the random variable $G_{i}$ from the scheme does not affect its strong security. However, this contradicts Lemma 2.4.4, as the resulting code has less than $\mu$ random symbols.

The following theorem proves a lower bound on the length of a $(\mu, t)$-strongly secure linear randomized index code.

Theorem 2.4.7. The length of a $\mu, t$ )-strongly secure linear $\eta$-randomized ( $m, n, \mathcal{X}$, $f)$-IC over $\mathbb{F}_{q}$ is at least $\kappa_{q}+\mu$. In other words, a linear randomized index code requires at least $\kappa_{q}+\mu$ transmissions in order to be $(\mu, t)$-strongly secure.

Proof. Suppose the linear randomized index code is based on $\boldsymbol{L}$. We divide the proof into several cases.

Case 1: $\eta=\mu$. Then, by Corollary 2.2.6 and Lemma 2.4.6, the subspace $\mathcal{C}(\boldsymbol{L})$ must contain:

1. the vectors $\boldsymbol{u}^{(i)}+\boldsymbol{e}_{f(i)}$ for some $\boldsymbol{u}^{(i)} \triangleleft \mathcal{X}_{i}$, for all $i \in[m]$;
2. the vectors $\boldsymbol{v}^{(i)}+\boldsymbol{e}_{n+i}$, for some $\boldsymbol{v}^{(i)} \triangleleft[n]$, for all $i \in[\mu]$.

Due to linear independence of these vectors and to the definition of $\kappa_{q}$, the length of the code is at least

$$
\begin{aligned}
\operatorname{dim}(\mathcal{C}(\boldsymbol{L})) & \geq \operatorname{rank}_{q}\left(\left\{\boldsymbol{u}^{(i)}+\boldsymbol{e}_{f(i)}\right\}_{i \in[m]}\right)+\operatorname{rank}_{q}\left(\left\{\boldsymbol{v}^{(i)}+\boldsymbol{e}_{n+i}\right\}_{i \in[\mu]}\right) \\
& \geq \kappa_{q}+\mu .
\end{aligned}
$$

Case 2: $\eta>\mu$, and for every $i \in[\eta]$ there exists some vector $\boldsymbol{v}^{(i)} \triangleleft[n]$ such that $\boldsymbol{v}^{(i)}+\boldsymbol{e}_{n+i} \in \mathcal{C}(\boldsymbol{L})$. In this case, similarly to Case 1, we have

$$
\operatorname{dim}(\mathcal{C}(\boldsymbol{L})) \geq \kappa_{q}+\eta>\kappa_{q}+\mu
$$

Therefore, $\boldsymbol{L}$ has at least $\kappa_{q}+\mu$ columns.

Case 3: $\eta>\mu$, and for some $i \in[\eta]$, it holds that $\boldsymbol{v}^{(i)}+\boldsymbol{e}_{n+i} \notin \mathcal{C}(\boldsymbol{L})$ for all $\boldsymbol{v}^{(i)} \triangleleft[n]$. By following exactly the same argument as in the proof of Lemma 2.4.6, we deduce that discarding $G_{i}$ does not affect the strong security of the randomized index code. By doing so, we obtain a new ( $\mu, t$ )-strongly secure randomized index code, which has $\eta-1$ random variables. This code is based on $\boldsymbol{L}^{\prime}$, which is obtained from $\boldsymbol{L}$ by deleting its $(n+i)$ th row.

The above argument can be applied until either the number of random variables decreases to $\mu$, or the code in consideration satisfies the condition of Case 2. In both cases, the resulting randomized index code has length at least $\kappa_{q}+\mu$. As the length of the code does not change during the process, we conclude that the length of the original code is at least $\kappa_{q}+\mu$.

### 2.4.2 A Construction of Optimal Strongly Secure Index Codes

In this subsection, we provide a construction of an optimal ( $\mu, t$ )-strongly secure $\mu$-randomized linear $(m, n, \mathcal{X}, f)$-IC over $\mathbb{F}_{q}$, which has length attaining the lower bound established in Theorem 2.4.7. This construction requires $q$ to be as large as $\kappa_{q}+\mu-1$. The proposed construction is based on the coset coding technique, originally introduced by Ozarow and Wyner [52]. This technique has been adopted in a variety of network coding applications $[12,26,29,59,71]$.

Construction A: Let $\boldsymbol{L}^{(0)}$ correspond to a linear $(m, n, \mathcal{X}, f)$-IC over $\mathbb{F}_{q}$ of optimal length $\kappa_{q}$. Let $\boldsymbol{M}$ be an invertible matrix of order $N=\kappa_{q}+\mu$, so that the last $\mu$ rows of $\boldsymbol{M}$ form a generator matrix of an MDS code. Let $\boldsymbol{P}$ be the submatrix of $\boldsymbol{M}$ formed by the first $\kappa_{q}$ rows, and $\boldsymbol{Q}$ the submatrix formed by the last $\mu$ rows
of $\boldsymbol{M}$. Let

$$
L=\left(\frac{L^{(0)} P}{Q}\right)
$$

Lemma 2.4.8. The matrix $\boldsymbol{L}$ in Construction $A$ corresponds to a $\mu$-randomized $(m, n, \mathcal{X}, f)$-IC over $\mathbb{F}_{q}$.

Proof. Let $\boldsymbol{g} \in \mathbb{F}_{q}^{\mu}$ be a random vector, uniformly distributed over $\mathbb{F}_{q}^{\mu}$. The encoding process is as follows

$$
\boldsymbol{x} \mapsto(\boldsymbol{x} \mid \boldsymbol{g}) \boldsymbol{L}=\boldsymbol{x} \boldsymbol{L}^{(0)} \boldsymbol{P}+\boldsymbol{g} \boldsymbol{Q}=\left(\boldsymbol{x} \boldsymbol{L}^{(0)} \mid \boldsymbol{g}\right) \boldsymbol{M}
$$

Since $\boldsymbol{M}$ is invertible, each receiver is able to recover $\left(\boldsymbol{x} \boldsymbol{L}^{(0)} \mid \boldsymbol{g}\right)$. Therefore, for each $i \in[m]$, the receiver $R_{i}$ can recover $\boldsymbol{x} \boldsymbol{L}^{(0)}$, and hence, can also recover $x_{f(i)}$, as $\boldsymbol{L}^{(0)}$ corresponds to a linear $(m, n, \mathcal{X}, f)$-IC over $\mathbb{F}_{q}$.

Lemma 2.4.9. The $\mu$-randomized ( $m, n, \mathcal{X}, f$ )-IC based on the matrix $\boldsymbol{L}$ in Construction $A$ is $(\mu, t)$-strongly secure.

Proof. Suppose that the adversary $A$ possess a message vector $\boldsymbol{x}_{\mathcal{X}_{A}}$, where $\left|\mathcal{X}_{A}\right|=t$. Additionally, $A$ can eavesdrop $\mu$ transmissions, that is, it has knowledge of $\boldsymbol{b} \triangleq$ $(\boldsymbol{x} \mid \boldsymbol{g}) \boldsymbol{L}[W]$, for some $W \subseteq[N]$ and $|W|=\mu$. Below, we show that the entropy of $\boldsymbol{X}_{\widehat{\mathcal{X}}_{A}}$ is not changed given the knowledge of $(\boldsymbol{X} \mid \boldsymbol{G}) \boldsymbol{L}[W]$ and of $\boldsymbol{x}_{\mathcal{X}_{A}}$. It suffices to show that for all $\boldsymbol{a} \in \mathbb{F}_{q}^{n-t}$ :

$$
\begin{equation*}
\operatorname{Pr}\left(\boldsymbol{X}_{\widehat{\mathcal{X}}_{A}}=\boldsymbol{a} \mid(\boldsymbol{X} \mid \boldsymbol{G}) \boldsymbol{L}[W]=\boldsymbol{b}, \boldsymbol{X}_{\mathcal{X}_{A}}=\boldsymbol{x}_{\mathcal{X}_{A}}\right)=\frac{1}{q^{n-t}} . \tag{2.19}
\end{equation*}
$$

The probability on the left-hand side of (2.19) can be rewritten as

$$
\begin{equation*}
\frac{\operatorname{Pr}\left(\boldsymbol{X}_{\widehat{\mathcal{X}}_{A}}=\boldsymbol{a},(\boldsymbol{X} \mid \boldsymbol{G}) \boldsymbol{L}[W]=\boldsymbol{b} \mid \boldsymbol{X}_{\mathcal{X}_{A}}=\boldsymbol{x}_{\mathcal{X}_{A}}\right)}{\operatorname{Pr}\left((\boldsymbol{X} \mid \boldsymbol{G}) \boldsymbol{L}[W]=\boldsymbol{b} \mid \boldsymbol{X}_{\mathcal{X}_{A}}=\boldsymbol{x}_{\mathcal{X}_{A}}\right)} \tag{2.20}
\end{equation*}
$$

The numerator in (2.20) is given by

$$
\begin{align*}
& \operatorname{Pr}\left(\boldsymbol{X}_{\widehat{\mathcal{X}}_{A}}=\boldsymbol{a},(\boldsymbol{X} \mid \boldsymbol{G}) \boldsymbol{L}[W]=\boldsymbol{b} \mid \boldsymbol{X}_{\mathcal{X}_{A}}=\boldsymbol{x}_{\mathcal{X}_{A}}\right) \\
& =\operatorname{Pr}\left(\boldsymbol{X}_{\widehat{\mathcal{X}}_{A}}=\boldsymbol{a} \mid \boldsymbol{X}_{\mathcal{X}_{A}}=\boldsymbol{x}_{\mathcal{X}_{A}}\right) \\
& \quad \times \operatorname{Pr}\left((\boldsymbol{X} \mid \boldsymbol{G}) \boldsymbol{L}[W]=\boldsymbol{b} \mid \boldsymbol{x}_{\widehat{\mathcal{X}}_{A}}=\boldsymbol{a}, \boldsymbol{X}_{\mathcal{X}_{A}}=\boldsymbol{x}_{\mathcal{X}_{A}}\right) \\
& =\frac{1}{q^{n-t}} \operatorname{Pr}\left((\boldsymbol{X} \mid \boldsymbol{G}) \boldsymbol{L}[W]=\boldsymbol{b} \mid \boldsymbol{X}_{\widehat{\mathcal{X}}_{A}}=\boldsymbol{a}, \boldsymbol{X}_{\mathcal{X}_{A}}=\boldsymbol{x}_{\mathcal{X}_{A}}\right)  \tag{2.21}\\
& =\frac{1}{q^{n-t}} \frac{1}{q^{\mu}} \\
& =\frac{1}{q^{n-t+\mu}} .
\end{align*}
$$

The penultimate transition can be explained as follows. We have

$$
\begin{equation*}
\boldsymbol{b}=(\boldsymbol{X} \mid \boldsymbol{G}) \boldsymbol{L}[W]=\boldsymbol{X} \boldsymbol{L}^{(0)} \boldsymbol{P}[W]+\boldsymbol{G} \boldsymbol{Q}[W] . \tag{2.22}
\end{equation*}
$$

The matrix $\boldsymbol{Q}[W]$ is invertible due to the fact that $\boldsymbol{Q}$ is a generator matrix of an $[N, \mu]$ MDS code. Since $\boldsymbol{X}$ is known, the system (2.22) has a unique solution given by

$$
\boldsymbol{G}=\left(\boldsymbol{b}-\boldsymbol{X} \boldsymbol{L}^{(0)} \boldsymbol{P}[W]\right)(\boldsymbol{Q}[W])^{-1}
$$

Since $\boldsymbol{G}$ is uniformly distributed over $\mathbb{F}_{q}^{\mu}$,

$$
\begin{aligned}
& \operatorname{Pr}\left((\boldsymbol{X} \mid \boldsymbol{G}) \boldsymbol{L}[W]=\boldsymbol{b} \mid \boldsymbol{X}_{\widehat{\mathcal{X}}_{A}}=\boldsymbol{a}, \boldsymbol{X}_{\mathcal{X}_{A}}=\boldsymbol{x}_{\mathcal{X}_{A}}\right) \\
& =\operatorname{Pr}\left(\boldsymbol{G}=\left(\boldsymbol{b}-\boldsymbol{X} \boldsymbol{L}^{(0)} \boldsymbol{P}[W]\right)(\boldsymbol{Q}[W])^{-1}\right) \\
& =\frac{1}{q^{\mu}} .
\end{aligned}
$$

Similarly to (2.21), the denominator in (2.20) is

$$
\begin{align*}
& \operatorname{Pr}\left((\boldsymbol{X} \mid \boldsymbol{G}) \boldsymbol{L}[W]=\boldsymbol{b} \mid \boldsymbol{X}_{\mathcal{X}_{A}}=\boldsymbol{x}_{\mathcal{X}_{A}}\right) \\
& =\sum_{\boldsymbol{c \in \mathbb { F } _ { q } ^ { n - t }}} \operatorname{Pr}\left(\boldsymbol{X}_{\widehat{\mathcal{X}}_{A}}=\boldsymbol{c} \mid \boldsymbol{X}_{\mathcal{X}_{A}}=\boldsymbol{x}_{\mathcal{X}_{A}}\right) \\
& \quad \times \operatorname{Pr}\left((\boldsymbol{X} \mid \boldsymbol{G}) \boldsymbol{L}[W]=\boldsymbol{b} \mid \boldsymbol{X}_{\widehat{\mathcal{X}}_{A}}=\boldsymbol{c}, \boldsymbol{X}_{\mathcal{X}_{A}}=\boldsymbol{x}_{\mathcal{X}_{A}}\right)  \tag{2.23}\\
& =q^{n-t} \frac{1}{q^{n-t}} \frac{1}{q^{\mu}} \\
& =\frac{1}{q^{\mu}} .
\end{align*}
$$

From (2.20), (2.21), and (2.23), we obtain (2.19), as claimed.

From Theorem 2.4.7, Lemma 2.4.8, and Lemma 2.4.9, we have the following theorem.

Theorem 2.4.10. The length of an optimal ( $\mu, t)$-strongly secure linear $\eta$-randomized $(m, n, \mathcal{X}, f)$-IC over $\mathbb{F}_{q}$ is at least $\kappa_{q}+\mu$. Moreover, the code based on the matrix $\boldsymbol{L}$ established in Construction $A$ achieves this bound $\left(q \geq \kappa_{q}+\mu-1\right)$.

## 3. INDEX CODE AND ERROR CORRECTION

Part of the work in this chapter was presented in the 2011 IEEE Symposium on Information Theory [21]. In this chapter, a generalization of index coding scheme, where transmitted symbols are subject to errors, is studied. Error-correcting methods for such a scheme, and their parameters, are investigated. In particular, the following question is discussed: given the side information hypergraph of index coding scheme and the maximal number of erroneous symbols $\delta$, what is the shortest (optimal) length of a linear index code which guarantees that every receiver is able to recover the required information? This question turns out to be a generalization of the problem of finding the shortest length of an error-correcting code with a prescribed error-correcting capability in the classical coding theory.

The Singleton bound and two other bounds, referred to as the $\alpha$-bound and the $\kappa$-bound, for the optimal length of a linear error-correcting index code (ECIC) are established. For large alphabets, a construction based on the concatenation of an optimal index code with an MDS classical code, is shown to attain the Singleton bound. For smaller alphabets, however, this construction may not be optimal. A random construction is also analyzed. It yields another implicit bound on the length of an optimal linear ECIC.

The problem of decoding a linear ECIC is studied. It is shown that in order to decode correctly the desired symbol, the decoder is required to find a vector that belongs to an affine space containing the actual error vector. The syndrome
decoding is shown to produce the correct output if the weight of the error pattern is less or equal to the error-correcting capability of the corresponding ECIC.

Furthermore, we introduce the notion of a static ECIC, which is suitable for use with a family of ICSI instances. Several bounds on the length of static ECIC's are derived, and constructions for static ECIC's are discussed. Connections of these codes to weakly resilient Boolean functions are also established.

Finally, we discuss the nonlinear ECIC's. Analogous bounds on the length of an optimal ECIC are established.

### 3.1 Error-Correcting Index Code with Side Information

Due to noise, the symbols received by $R_{i}, i \in[m]$, may be subject to errors. Consider an ICSI instance $(m, n, \mathcal{X}, f)$, and assume that $S$ broadcasts a vector $\mathfrak{E}(\boldsymbol{x}) \in \mathbb{F}_{q}^{N}$. Let $\boldsymbol{\epsilon}^{(i)} \in \mathbb{F}_{q}^{N}$ be the error affecting the information received by $R_{i}, i \in[m]$. Then $R_{i}$ actually receives the vector

$$
\boldsymbol{y}_{i}=\mathfrak{E}(\boldsymbol{x})+\boldsymbol{\epsilon}^{(i)} \in \mathbb{F}_{q}^{N},
$$

instead of $\mathfrak{E}(\boldsymbol{x})$. The following definition is a generalization of Definition 2.1.3.

Definition 3.1.1. Consider an ICSI instance described by $\mathcal{H}=\mathcal{H}(m, n, \mathcal{X}, f)$. A $\delta$-error-correcting index code ( $\delta$-error-correcting $\mathcal{H}$-IC, or simpler, $(\delta, \mathcal{H})$-ECIC) over $\mathbb{F}_{q}$ for this instance is an encoding function

$$
\mathfrak{E}: \mathbb{F}_{q}^{n} \rightarrow \mathbb{F}_{q}^{N}
$$

such that for each receiver $R_{i}, i \in[m]$, there exists a decoding function

$$
\mathfrak{D}_{i}: \mathbb{F}_{q}^{N} \times \mathbb{F}_{q}^{\left|\mathcal{X}_{i}\right|} \rightarrow \mathbb{F}_{q},
$$

satisfying

$$
\forall \boldsymbol{x}, \boldsymbol{\epsilon}^{(i)} \in \mathbb{F}_{q}^{n}, \operatorname{wt}\left(\boldsymbol{\epsilon}^{(i)}\right) \leq \delta: \mathfrak{D}_{i}\left(\mathfrak{E}(\boldsymbol{x})+\boldsymbol{\epsilon}^{(i)}, \boldsymbol{x}_{\mathcal{X}_{i}}\right)=x_{f(i)} .
$$

The definitions of the length, of a linear index code, and of the matrix corresponding to a linear index code are naturally extended to an error-correcting index code. Note that if $\mathfrak{E}$ is an $\mathcal{H}$-IC over $\mathbb{F}_{q}$, then it is a $(0, \mathcal{H})$-ECIC, and vice versa.

Definition 3.1.2. An optimal linear $(\delta, \mathcal{H})$-ECIC over $\mathbb{F}_{q}$ is a linear $(\delta, \mathcal{H})$-ECIC over $\mathbb{F}_{q}$ of the smallest possible length $\mathcal{N}_{q}[\delta, \mathcal{H}]$.

Consider an ICSI instance described by $\mathcal{H}=\mathcal{H}(m, n, \mathcal{X}, f)$. We define the set of vectors

$$
\mathcal{I}(q, \mathcal{H}) \triangleq\left\{\boldsymbol{z} \in \mathbb{F}_{q}^{n}: \exists i \in[m] \text { such that } \boldsymbol{z}_{\mathcal{X}_{i}}=\mathbf{0} \text { and } z_{f(i)} \neq 0\right\}
$$

For all $i \in[m]$, we also define

$$
\mathcal{Y}_{i} \triangleq[n] \backslash\left(\{f(i)\} \cup \mathcal{X}_{i}\right) .
$$

Then the collection of supports of all vectors in $\mathcal{I}(q, \mathcal{H})$ is given by

$$
\begin{equation*}
\mathcal{J}(\mathcal{H}) \triangleq \bigcup_{i \in[m]}\left\{\{f(i)\} \cup Y_{i}: Y_{i} \subseteq \mathcal{Y}_{i}\right\} \tag{3.1}
\end{equation*}
$$

The necessary and sufficient condition for a matrix $\boldsymbol{L}$ to be the matrix corresponding to some $(\delta, \mathcal{H})$-ECIC is given in the following lemma.

Lemma 3.1.3. The matrix $\boldsymbol{L}$ corresponds to a $(\delta, \mathcal{H})$-ECIC over $\mathbb{F}_{q}$ if and only if

$$
\begin{equation*}
\text { wt }(\boldsymbol{z} \boldsymbol{L}) \geq 2 \delta+1 \text { for all } \boldsymbol{z} \in \mathcal{I}(q, \mathcal{H}) \tag{3.2}
\end{equation*}
$$

Equivalently, L corresponds to a $(\delta, \mathcal{H})$-ECIC over $\mathbb{F}_{q}$ if and only if

$$
\text { wt }\left(\sum_{i \in K} z_{i} \boldsymbol{L}_{i}\right) \geq 2 \delta+1
$$

for all $K \in \mathcal{J}(\mathcal{H})$ and for all choices of $z_{i} \in \mathbb{F}_{q}^{*}, i \in K$.

Proof. For each $\boldsymbol{x} \in \mathbb{F}_{q}^{n}$, we define

$$
B(\boldsymbol{x}, \delta)=\left\{\boldsymbol{y} \in \mathbb{F}_{q}^{N}: \boldsymbol{y}=\boldsymbol{x} \boldsymbol{L}+\boldsymbol{\epsilon}, \boldsymbol{\epsilon} \in \mathbb{F}_{q}^{N}, \mathrm{wt}(\boldsymbol{\epsilon}) \leq \delta\right\},
$$

the set of all vectors resulting from at most $\delta$ errors in the transmitted vector associated with the information vector $\boldsymbol{x}$. Then the receiver $R_{i}$ can recover $x_{f(i)}$ correctly if and only if

$$
B(\boldsymbol{x}, \delta) \cap B\left(\boldsymbol{x}^{\prime}, \delta\right)=\varnothing
$$

for every pair $\boldsymbol{x}, \boldsymbol{x}^{\prime} \in \mathbb{F}_{q}^{n}$ satisfying:

$$
\boldsymbol{x}_{\mathcal{X}_{i}}=\boldsymbol{x}_{\mathcal{X}_{i}}^{\prime} \text { and } x_{f(i)} \neq x_{f(i)}^{\prime}
$$

(Recall that $R_{i}$ is interested only in the bit $x_{f(i)}$, not in the whole vector $\boldsymbol{x}$.)
Therefore, $\boldsymbol{L}$ corresponds to a $(\delta, \mathcal{H})$-ECIC if and only if the following condition is satisfied: for all $i \in[m]$ and for all $\boldsymbol{x}, \boldsymbol{x}^{\prime} \in \mathbb{F}_{q}^{n}$ such that $\boldsymbol{x}_{\mathcal{X}_{i}}=\boldsymbol{x}_{\mathcal{X}_{i}}^{\prime}$ and $x_{f(i)} \neq x_{f(i)}^{\prime}$,
it holds

$$
\begin{equation*}
\forall \boldsymbol{\epsilon}, \boldsymbol{\epsilon}^{\prime} \in \mathbb{F}_{q}^{N}, \mathrm{wt}(\boldsymbol{\epsilon}) \leq \delta, \mathrm{wt}\left(\boldsymbol{\epsilon}^{\prime}\right) \leq \delta: \boldsymbol{x} \boldsymbol{L}+\boldsymbol{\epsilon} \neq \boldsymbol{x}^{\prime} \boldsymbol{L}+\boldsymbol{\epsilon}^{\prime} \tag{3.3}
\end{equation*}
$$

Denote $\boldsymbol{z}=\boldsymbol{x}^{\prime}-\boldsymbol{x}$. Then, the condition in (3.3) can be reformulated as follows: for all $i \in[n]$ and for all $\boldsymbol{z} \in \mathbb{F}_{q}^{n}$ such that $\boldsymbol{z}_{\mathcal{X}_{i}}=\mathbf{0}$ and $z_{f(i)} \neq 0$, it holds

$$
\begin{equation*}
\forall \boldsymbol{\epsilon}, \boldsymbol{\epsilon}^{\prime} \in \mathbb{F}_{q}^{N}, \mathrm{wt}(\boldsymbol{\epsilon}) \leq \delta, \mathrm{wt}\left(\boldsymbol{\epsilon}^{\prime}\right) \leq \delta: \boldsymbol{z} \boldsymbol{L} \neq \boldsymbol{\epsilon}-\boldsymbol{\epsilon}^{\prime} \tag{3.4}
\end{equation*}
$$

An equivalent condition is that for all $\boldsymbol{z} \in \mathcal{I}(q, \mathcal{H})$,

$$
\mathrm{wt}(\boldsymbol{z} \boldsymbol{L}) \geq 2 \delta+1
$$

Since for $\boldsymbol{z} \in \mathcal{I}(q, \mathcal{H})$ we have

$$
\boldsymbol{z} \boldsymbol{L}=\sum_{i \in \operatorname{supp}(\boldsymbol{z})} z_{i} \boldsymbol{L}_{i},
$$

the condition (3.2) can be restated as

$$
\mathrm{wt}\left(\sum_{i \in K} z_{i} \boldsymbol{L}_{i}\right) \geq 2 \delta+1
$$

for all $K \in \mathcal{J}(\mathcal{H})$ and for all choices of nonzero $z_{i} \in \mathbb{F}_{q}, i \in K$.

The next corollary follows from Lemma 3.1.3 in a straightforward manner. It is not hard to see that the conditions stated in Lemma 3.1.3 and in the corollary below are, in fact, equivalent.

Corollary 3.1.4. For all $i \in[m]$, let

$$
\boldsymbol{M}_{i} \triangleq \operatorname{span}_{q}\left(\left\{\boldsymbol{L}_{j}: j \in \mathcal{Y}_{i}\right\}\right)
$$

Then, the matrix $\boldsymbol{L}$ corresponds to a $(\delta, \mathcal{H})$-ECIC over $\mathbb{F}_{q}$ if and only if

$$
\begin{equation*}
\forall i \in[m]: \mathrm{d}\left(\boldsymbol{L}_{f(i)}, \boldsymbol{M}_{i}\right) \geq 2 \delta+1 \tag{3.5}
\end{equation*}
$$

The next corollary also follows directly from Lemma 3.1 .3 by considering an error-free setup, that is, $\delta=0$.

Corollary 3.1.5. The matrix $\boldsymbol{L}$ corresponds to an $\mathcal{H}$-IC over $\mathbb{F}_{q}$ if and only if

$$
\mathrm{wt}\left(\sum_{i \in K} z_{i} \boldsymbol{L}_{i}\right) \geq 1
$$

for all $K \in \mathcal{J}(\mathcal{H})$ and for all choices of $z_{i} \in \mathbb{F}_{q}^{*}, i \in K$, or, equivalently,

$$
\forall i \in[m]: \boldsymbol{L}_{f(i)} \notin \operatorname{span}_{q}\left(\left\{\boldsymbol{L}_{j}\right\}_{j \in \mathcal{Y}_{i}}\right)
$$

Remark 3.1.6. The conditions stated in Corollary 3.1.5 and Corollary 2.2.6 are, as expected, equivalent. Indeed, for each $i \in[m]$ we have

$$
\begin{aligned}
\boldsymbol{L}_{f(i)} \notin \operatorname{span}_{q}\left(\left\{\boldsymbol{L}_{j}\right\}_{j \in \mathcal{Y}_{i}}\right) & \Longleftrightarrow \exists \boldsymbol{v}^{(i)}: \boldsymbol{v}^{(i)} \boldsymbol{L}_{f(i)}^{\mathrm{T}}=1 \text { and } \boldsymbol{v}^{(i)} \boldsymbol{L}_{\mathcal{Y}_{i}}^{\mathrm{T}}=\mathbf{0} \\
& \Longleftrightarrow \exists \boldsymbol{v}^{(i)}: \boldsymbol{c}^{(i)}=\boldsymbol{v}^{(i)} \boldsymbol{L}^{\mathrm{T}} \in \mathcal{C}(\boldsymbol{L}) \text { satisfies } \boldsymbol{c}_{f(i)}^{(i)}=1 \text { and } \\
& \boldsymbol{c}_{\mathcal{Y}_{i}}^{(i)}=\mathbf{0} \\
& \Longleftrightarrow \exists \boldsymbol{u}^{(i)} \in \mathbb{F}_{q}^{n}: \boldsymbol{u}^{(i)} \triangleleft \mathcal{X}_{i} \text { and } \boldsymbol{c}^{(i)}=\boldsymbol{u}^{(i)}+\boldsymbol{e}_{f(i)} \in \mathcal{C}(\boldsymbol{L}) .
\end{aligned}
$$

Example 3.1.7. Let $q=2, m=n=3$, and $f(i)=i$ for $i \in$ [3]. Suppose
$\mathcal{X}_{1}=\{2,3\}, \mathcal{X}_{2}=\{1,3\}$, and $\mathcal{X}_{3}=\{1,2\}$. Let

$$
\boldsymbol{L}=\left(\begin{array}{llll}
1 & 1 & 1 & 0 \\
1 & 1 & 0 & 1 \\
1 & 0 & 1 & 1
\end{array}\right)
$$

Note that the rows of $\boldsymbol{L}$ generate a $[4,3,1]_{2}$ code, which has minimum distance one. Nevertheless, the index code based on $\boldsymbol{L}$ can still correct one error. Indeed, let $\mathcal{H}=\mathcal{H}(3,3, \mathcal{X}, f)$, we have

$$
\mathcal{I}(2, \mathcal{H})=\{100,010,001\}
$$

Since each row of $\boldsymbol{L}$ has weight at least three, it follows that $\mathbf{w t}(\boldsymbol{z} \boldsymbol{L}) \geq 3$ for all $\boldsymbol{z} \in \mathcal{I}(2, \mathcal{H})$. By Lemma 3.1.3, $\boldsymbol{L}$ corresponds to a $(1, \mathcal{H})$-ECIC over $\mathbb{F}_{2}$.

In fact, for this instance, even a simpler index code of length three, based on

$$
\boldsymbol{L}^{\prime}=\left(\begin{array}{lll}
1 & 1 & 1 \\
1 & 1 & 1 \\
1 & 1 & 1
\end{array}\right)
$$

is a $(1, \mathcal{H})$-ECIC over $\mathbb{F}_{2}$.

Example 3.1.8. Assume that $m=n$ and $f(i)=i$ for all $i \in[m]$. Furthermore, suppose that $\mathcal{X}_{i}=\varnothing$ for all $i \in[m]$ (that is, there is no side information available to the receivers). Let $\mathcal{H}=\mathcal{H}(m, n, \mathcal{X}, f)$. Then, $\mathcal{I}(q, \mathcal{H})=\mathbb{F}_{q}^{n} \backslash\{0\}$. Hence, by Lemma 3.1.3, the $n \times N$ matrix $\boldsymbol{L}$ corresponding to a $(\delta, \mathcal{H})$-ECIC over $\mathbb{F}_{q}$ (for some integer $\delta \geq 0$ ) if and only if it is a generating matrix of an $[N, n, \geq 2 \delta+1]_{q}$ linear code. Thus, under these settings, the problem of designing an optimal linear ECIC
is reduced to the problem of constructing an optimal classical linear error-correcting code.

Observe however, that for general $\mathcal{X}$, changing the order of rows in $\boldsymbol{L}$ can lead to ECIC's with different error-correcting capabilities. Therefore, the problem of designing an optimal linear ECIC is essentially the problem of finding the matrix $\boldsymbol{L}$ corresponding to that code. However, the minimum distance of the code generated by the rows of $\boldsymbol{L}$ is not necessary a valid indicator for goodness of an ECIC. Sometimes, as Example 3.1.7 shows, matrix $\boldsymbol{L}$ with redundant rows yields a good ECIC.

### 3.2 The $\alpha$-Bound and the $\kappa$-Bound

Let ( $m, n, \mathcal{X}, f$ ) be an ICSI instance, and let $\mathcal{H}$ be the corresponding side information hypergraph. Next, we introduce the following definitions for the hypergraph $\mathcal{H}$.

Definition 3.2.1. A subset $H$ of $[n]$ is called a generalized independent set in $\mathcal{H}$ if every nonempty subset $K$ of $H$ belongs to $\mathcal{J}(\mathcal{H})$.

Definition 3.2.2. A generalized independent set of maximum size in $\mathcal{H}$ is called a maximum generalized independent set. The size of a maximum generalized independent set in $\mathcal{H}$ is called the generalized independence number, and denoted by $\alpha(\mathcal{H})$.

In the following two lemmas, we consider an ICSI instance ( $m, n, \mathcal{X}, f$ ) in which $m=n$ and $f(i)=i$ for all $i \in[n]$. Let $\mathcal{D}$ be the underlying digraph of the side information hypergraph $\mathcal{H}$, that is, $\mathcal{D}$ is the side information digraph of this ICSI instance (see Remark 2.1.2). A maximum acyclic induced subgraph of $\mathcal{D}$ is an acyclic
subgraph which is induced by a set of maximum number of vertices. We use $\alpha(\mathcal{D})$ to denote the order of a maximum acyclic induced subgraph of $\mathcal{D}$. The two lemmas below show that $\alpha(\mathcal{H})$ is indeed a generalization of $\alpha(\mathcal{D})$ and $\alpha(\mathcal{G})$.

Lemma 3.2.3. $\alpha(\mathcal{H})=\alpha(\mathcal{D})$.

Proof. It suffices to show that a generalized independent set in $\mathcal{H}$ induces an acyclic subgraph of $\mathcal{D}$ and vice versa.

Let $H$ be a generalized independent set in $\mathcal{H}$. If $|H|=1$, then obviously $H$ induces an acyclic subgraph of $\mathcal{D}$. Assume that $|H| \geq 2$. Take any subset $K$ of $H$ where $|K| \geq 2$. Since $K \in \mathcal{J}(\mathcal{H})$, there exists some $i \in[n]$ such that $i=f(i) \in K$ and $j \in \mathcal{Y}_{i}$ for all $j \in K \backslash\{i\}$. Recall that $\mathcal{Y}_{i}=[n] \backslash\left(\{i\} \cup \mathcal{X}_{i}\right)$. As $(i, j)$ 's for $j \in \mathcal{X}_{i}$ are the only arcs of $\mathcal{D}$ that originate from $i$, we deduce that $\mathcal{D}$ has no arc of the form $(i, j)$ for every $j \in K \backslash\{i\}$. Therefore, $K$ cannot form a circuit in $\mathcal{D}$. Since this conclusion holds for any such $K$, we conclude that $H$ induces an acyclic subgraph of $\mathcal{D}$.

Conversely, suppose that $H$ induces an acyclic subgraph of $\mathcal{D}$. We aim to show that $H$ is a generalized independent set in $\mathcal{H}$. Take an arbitrary nonempty subset $K$ of $H$ and let $\mathcal{D}_{K}$ be the $K$-induced subgraph of $\mathcal{D}$. If $K=\{i\}$ for some $i$, then obviously $K \in \mathcal{J}(\mathcal{H})$. Suppose $|K| \geq 2$. Since $\mathcal{D}_{K}$ is also acyclic, there exists some vertex $i \in K$ with no out-going arcs in $\mathcal{D}_{K}$. In other words, $(i, j) \notin \mathcal{E}\left(\mathcal{D}_{K}\right)$ for every $j \in K \backslash\{i\}$. Therefore, $j \in \mathcal{Y}_{i}$ for every $j \in K \backslash\{i\}$, which implies that $K \backslash\{i\} \subseteq \mathcal{Y}_{i}$. Hence $K=\{f(i)=i\} \cup(K \backslash\{i\}) \in \mathcal{J}(\mathcal{H})$. Since $K$ is an arbitrary nonempty subset of $H$, we conclude that $H$ is a generalized independent set of $\mathcal{H}$.

Lemma 3.2.4. If $\mathcal{D}$ is symmetric, then $\alpha(\mathcal{H})=\alpha(\mathcal{G})$.

Proof. It suffices to show that if $\mathcal{D}$ is symmetric, then the set of generalized inde-
pendent sets of $\mathcal{H}$ and the set of independent sets of $\mathcal{G}$ coincide.
Let $H$ be a generalized independent set in $\mathcal{H}$. If $|\mathcal{H}|=1$, then obviously $H$ is an independent set in $\mathcal{G}$. Assume that $|H| \geq 2$. For any pair of vertices $i, j$ in $H$, the set $\{i, j\}$ belongs to $\mathcal{J}(\mathcal{H})$. By definition of $\mathcal{J}(\mathcal{H})$, either $(i, j) \notin \mathcal{E}(\mathcal{D})$ or $(j, i) \notin \mathcal{E}(\mathcal{D})$. Since $\mathcal{D}$ is symmetric, there are no arcs between $i$ and $j$, in neither directions. Therefore, $H$ is an independent set in $\mathcal{G}$.

Conversely, let $H$ be an independent set in $\mathcal{G}$. For each $i \in H$, since there are no $\operatorname{arcs}$ in $\mathcal{D}$ from $i$ to any of the other vertices in $H$, we deduce that $H \backslash\{i\} \subseteq \mathcal{Y}_{i}$. Due to (3.1), every subset of $H$ that contains $i$ belongs to $\mathcal{J}(\mathcal{H})$. This holds for an arbitrary $i \in H$. Therefore, every nonempty subset of $H$ belongs to $\mathcal{J}(\mathcal{H})$. Hence $H$ is a generalized independent set of $\mathcal{H}$.

Next, we present a lower bound on the length of a $(\delta, \mathcal{H})$-ECIC over $\mathbb{F}_{q}$. We call this bound the $\alpha$-bound.

Theorem 3.2.5 ( $\alpha$-bound). The length of an optimal linear $(\delta, \mathcal{H})$-ECIC over $\mathbb{F}_{q}$ satisfies

$$
\mathcal{N}_{q}[\delta, \mathcal{H}] \geq N_{q}[\alpha(\mathcal{H}), 2 \delta+1] .
$$

Proof. Consider an $n \times N$ matrix $L$, which corresponds to a $(\delta, \mathcal{H})$-ECIC over $\mathbb{F}_{q}$. Let $H=\left\{i_{1}, i_{2}, \ldots, i_{\alpha(\mathcal{H})}\right\}$ be a maximum generalized independent set in $\mathcal{H}$. Then, every subset $K \subseteq H$ satisfies $K \in \mathcal{J}(\mathcal{H})$. Therefore,

$$
\mathrm{wt}\left(\sum_{i \in K} z_{i} \boldsymbol{L}_{i}\right) \geq 2 \delta+1
$$

for all $K \subseteq H$, where $K \neq \varnothing$, and for all choices of $z_{i} \in \mathbb{F}_{q}^{*}, i \in K$. Hence, the $\alpha(\mathcal{H})$ rows of $\boldsymbol{L}$, namely $\boldsymbol{L}_{i_{1}}, \boldsymbol{L}_{i_{2}}, \ldots, \boldsymbol{L}_{i_{\alpha(\mathcal{H})}}$, form a generator matrix of an
$[N, \alpha(\mathcal{H}), 2 \delta+1]_{q}$ code. Therefore,

$$
N \geq N_{q}[\alpha(\mathcal{H}), 2 \delta+1] .
$$

Example 3.2.6. Let $q=2, m=n=5, f(i)=i$ for all $i \in[m]$, and $\delta=2$. Assume

$$
\mathcal{X}_{1}=\{2,3,4\}, \mathcal{X}_{2}=\{3,4,5\}, \mathcal{X}_{3}=\{4,5,1\}, \mathcal{X}_{4}=\{5,1,2\}, \mathcal{X}_{5}=\{1,2,3\}
$$

Let $\mathcal{H}=\mathcal{H}(5,5, \mathcal{X}, f)$. Then

$$
\mathcal{J}(\mathcal{H})=\{\{1\},\{1,5\},\{2\},\{2,1\},\{3\},\{3,2\},\{4\},\{4,3\},\{5\},\{5,4\}\}
$$

It is easy to check that $\alpha(\mathcal{H})=2$. Therefore, Theorem 3.2.5 implies that

$$
\mathcal{N}_{2}[2, \mathcal{H}] \geq N_{2}[2,5]=8
$$

The last equality can be verified by [31].

Remark 3.2.7. In [3], when $m=n$ and $f(i)=i$ for all $i \in[n]$, the quantity $\alpha(\mathcal{D})=\alpha(\mathcal{H})$ was shown to be a lower bound for the length of a linear (non-errorcorrecting) index code. However, the $\alpha$-bound in Theorem 3.2.5 does not follow from the results in [3]. The reason is that the concatenation of an optimal linear errorcorrecting code with an optimal linear (non-error-correcting) index code might fail to produce an optimal linear ECIC. This fact is illustrated later in Example 3.2.10.

The following proposition is based on the fact that concatenation of a $\delta$-errorcorrecting code with an optimal linear (non-error-correcting) $\mathcal{H}$-IC over $\mathbb{F}_{q}$ yields a linear $(\delta, \mathcal{H})$-ECIC over $\mathbb{F}_{q}$.

Proposition 3.2.8 ( $\kappa$-bound). The length of an optimal $(\delta, \mathcal{H})$-ECIC over $\mathbb{F}_{q}$ satisfies

$$
\mathcal{N}_{q}[\delta, \mathcal{H}] \leq N_{q}\left[\kappa_{q}(\mathcal{H}), 2 \delta+1\right] .
$$

Proof. Let $\boldsymbol{G}$, which is an $n \times \kappa_{q}(\mathcal{H})$ matrix, correspond to an optimal $\mathcal{H}$-IC over $\mathbb{F}_{q}$. Denote

$$
\boldsymbol{y}=\boldsymbol{x} \boldsymbol{G} \in \mathbb{F}_{q}^{\kappa_{q}(\mathcal{H})}
$$

Let $\boldsymbol{M}$ be a generator matrix of an optimal $\left[N, \kappa_{q}(\mathcal{H}), 2 \delta+1\right]_{q}$ code $\mathscr{C}$, where

$$
N=N_{q}\left[\kappa_{q}(\mathcal{H}), 2 \delta+1\right] .
$$

Consider a scheme where $S$ broadcasts the vector $\boldsymbol{y} \boldsymbol{M} \in \mathbb{F}_{q}^{N}$. If less than $\delta$ errors occur, then each receiver $R_{i}$ is able to recover $\boldsymbol{y}$ by using $\mathscr{C}$. Hence each $R_{i}$ is able to recover $x_{f(i)}$. Therefore, for the index code based on $\boldsymbol{L}$,

$$
L=\boldsymbol{G} \boldsymbol{M}
$$

each receiver $R_{i}$ is capable to recover $x_{f(i)}$ if the number of errors is at most $\delta$. The length of the corresponding ECIC is $N=N_{q}\left[\kappa_{q}(\mathcal{H}), 2 \delta+1\right]$. Therefore,

$$
\mathcal{N}_{q}[\delta, \mathcal{H}] \leq N_{q}\left[\kappa_{q}(\mathcal{H}), 2 \delta+1\right] .
$$

By combining the results in Theorem 3.2.5 and in Proposition 3.2.8, we obtain the following corollary.

Corollary 3.2.9. The length of an optimal linear $(\delta, \mathcal{H})$-ECIC over $\mathbb{F}_{q}$ satisfies

$$
N_{q}[\alpha(\mathcal{H}), 2 \delta+1] \leq \mathcal{N}_{q}[\delta, \mathcal{H}] \leq N_{q}\left[\kappa_{q}(\mathcal{H}), 2 \delta+1\right]
$$

It is shown in the example below that the inequalities in Corollary 3.2.9 can be strict. In particular, it follows that mere application of an error-correcting code on top of an index code may fail to provide us with an optimal linear ECIC. This fact motivates the study of ECIC's.

Example 3.2.10. Let $q=2, m=n=5, \delta=2$, and $f(i)=i$ for all $i \in[m]$. Assume

$$
\mathcal{X}_{1}=\{2,5\}, \mathcal{X}_{2}=\{1,3\}, \mathcal{X}_{3}=\{2,4\}, \mathcal{X}_{4}=\{3,5\}, \mathcal{X}_{5}=\{1,4\}
$$

Let $\mathcal{H}=\mathcal{H}(5,5, \mathcal{X}, f)$. Then we have

$$
\begin{aligned}
\mathcal{J}(\mathcal{H})=\{ & \{1\},\{1,3\},\{1,4\},\{1,3,4\} \\
& \{2\},\{2,4\},\{2,5\},\{2,4,5\} \\
& \{3\},\{1,3\},\{3,5\},\{1,3,5\} \\
& \{4\},\{1,4\},\{2,4\},\{1,2,4\} \\
& \{5\},\{2,5\},\{3,5\},\{2,3,5\}\}
\end{aligned}
$$

The side information graph $\mathcal{G}$ of this instance is a pentagon. It is easy to verify that $\alpha(\mathcal{H})=\alpha(\mathcal{G})=2$. It follows from [4, Theorem 9] that $\kappa_{2}(\mathcal{H})=\operatorname{minrk}_{2}(\mathcal{G})=3$. From [31] we have

$$
N_{2}[2,5]=8 \quad \text { and } \quad N_{2}[3,5]=10
$$

Due to Corollary 3.2.9, we have

$$
8 \leq \mathcal{N}_{2}[2, \mathcal{H}] \leq 10
$$

Using a computer search, we obtain that $\mathcal{N}_{2}[2, \mathcal{H}]=9$, and the corresponding optimal scheme is based on

$$
\boldsymbol{L}=\left(\begin{array}{lllllllll}
1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 \\
1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\
0 & 1 & 1 & 0 & 0 & 1 & 0 & 1 & 1 \\
1 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 1
\end{array}\right)
$$

It is technical to verify that for all $K \in \mathcal{J}(\mathcal{H})$,

$$
\mathrm{wt}\left(\sum_{i \in K} \boldsymbol{L}_{i}\right) \geq 5
$$

Therefore by Lemma 3.1.3, for the index code based on $\boldsymbol{L}$, each receiver $R_{i}$ is able to recover $x_{i}$, if the number of errors is less than or equal to 2 . Observe that the length of the ECIC corresponding to $\boldsymbol{L}$ lies strictly between the $\alpha$-bound and the $\kappa$-bound.

The following bounds on the minranks of graphs are known to hold [34]. Note that the second inequality in the theorem below is also called the clique-cover bound for minranks, as $\chi(\overline{\mathcal{G}})$ is nothing other than the clique cover number of $\mathcal{G}$.

Theorem 3.2.11 (Sandwich Theorem). Let $\chi(\overline{\mathcal{G}})$ denote the chromatic number of
the complement of the graph $\mathcal{G}$. Then,

$$
\alpha(\mathcal{G}) \leq \operatorname{minrk}_{q}(\mathcal{G}) \leq \chi(\overline{\mathcal{G}})
$$

Consider the basic case when $m=n$ and $f(i)=i$ for all $i \in[m]$, and the side information digraph $\mathcal{D}$ is symmetric. We have $\alpha(\mathcal{H})=\alpha(\mathcal{G})$ (Lemma 3.2.4) and $\kappa_{q}(\mathcal{H})=\operatorname{minrk}_{q}(\mathcal{G})$ (see the discussion that follows Corollary 2.2.8), where $\mathcal{G}$ is the side information graph (the underlying graph of $\mathcal{H}$ ). If $\mathcal{G}$ satisfies $\alpha(\mathcal{G})=\chi(\overline{\mathcal{G}})$, then

$$
\alpha(\mathcal{H})=\alpha(\mathcal{G})=\operatorname{minrk}_{q}(\mathcal{G})=\kappa_{q}(\mathcal{H}) .
$$

Hence from Corollary 3.2.9 we have

$$
\mathcal{N}_{q}[\delta, \mathcal{H}]=N_{q}[\alpha(\mathcal{H}), 2 \delta+1]=N_{q}\left[\kappa_{q}(\mathcal{H}), 2 \delta+1\right],
$$

for all $q$, and the corresponding bounds in Corollary 3.2.9 are tight.

Definition 3.2.12. A graph $\mathcal{G}$ is called perfect if for every induced subgraph $\mathcal{G}^{\prime}$ of $\mathcal{G}$, it holds that $\alpha\left(\mathcal{G}^{\prime}\right)=\chi\left(\overline{\mathcal{G}^{\prime}}\right)$.

Perfect graphs include families of graphs such as trees, bipartite graphs, interval graphs, and chordal graphs. If the side information graph $\mathcal{G}$ is perfect, then the bounds in Corollary 3.2.9 are tight. For the full characterization of perfect graphs, the reader can refer to [18].

### 3.3 The Singleton Bound

The following bound generalizes the Singleton bound for classical linear error-correcting codes to linear ECIC's.

Theorem 3.3.1 (Singleton bound). The length of an optimal linear $(\delta, \mathcal{H})$-ECIC over $\mathbb{F}_{q}$ satisfies

$$
\mathcal{N}_{q}[\delta, \mathcal{H}] \geq \kappa_{q}(\mathcal{H})+2 \delta .
$$

Proof. Let $\boldsymbol{L}$ be the $n \times \mathcal{N}_{q}[\delta, \mathcal{H}]$ matrix corresponding to some optimal ( $\delta, \mathcal{H}$ )-ECIC over $\mathbb{F}_{q}$. Let $\boldsymbol{L}^{\prime}$ be the matrix obtained by deleting any $2 \delta$ columns from $\boldsymbol{L}$.

By Lemma 3.1.3, L satisfies

$$
\mathrm{wt}\left(\sum_{i \in K} z_{i} \boldsymbol{L}_{i}\right) \geq 2 \delta+1
$$

for all $K \in \mathcal{J}(\mathcal{H})$ and all choices of $z_{i} \in \mathbb{F}_{q}^{*}, i \in K$. We deduce that the rows of $\boldsymbol{L}^{\prime}$ also satisfy

$$
\mathrm{wt}\left(\sum_{i \in K} z_{i} \boldsymbol{L}_{i}^{\prime}\right) \geq 1
$$

for all such $K$ and $z_{i}$ 's. By Corollary 3.1.5, $\boldsymbol{L}^{\prime}$ corresponds to a linear $\mathcal{H}$-IC over $\mathbb{F}_{q}$. Therefore, by Corollary $2.2 .8, \boldsymbol{L}^{\prime}$ has at least $\kappa_{q}(\mathcal{H})$ columns. We deduce that

$$
\mathcal{N}_{q}[\delta, \mathcal{H}]-2 \delta \geq \kappa_{q}(\mathcal{H})
$$

which concludes the proof.

The following corollary from Proposition 3.2.8 and Theorem 3.3.1 demonstrates that, for sufficiently large alphabets, a concatenation of a classical MDS errorcorrecting code with an optimal (non-error-correcting) index code yields an optimal
linear ECIC. On the contrary, as it is illustrated in Example 3.2.10, this does not hold for the index coding schemes over small alphabets.

Corollary 3.3.2 (MDS error-correcting index code). For $q \geq \kappa_{q}(\mathcal{H})+2 \delta-1$,

$$
\begin{equation*}
\mathcal{N}_{q}[\delta, \mathcal{H}]=\kappa_{q}(\mathcal{H})+2 \delta . \tag{3.6}
\end{equation*}
$$

Proof. From Theorem 3.3.1, we have

$$
\mathcal{N}_{q}[\delta, \mathcal{H}] \geq \kappa_{q}(\mathcal{H})+2 \delta
$$

On the other hand, from Proposition 3.2.8,

$$
\mathcal{N}_{q}[\delta, \mathcal{H}] \leq N_{q}\left[\kappa_{q}(\mathcal{H}), 2 \delta+1\right]=\kappa_{q}(\mathcal{H})+2 \delta,
$$

for $q \geq \kappa_{q}(\mathcal{H})+2 \delta-1$ (by taking doubly-extended Reed-Solomon codes). Therefore, for these $q$, (3.6) holds.

Remark 3.3.3. Let $q=2, m=n=2 \ell+1(\ell \geq 2)$, and $f(i)=i$ for all $i \in[n]$. Let $\mathcal{X}_{1}=\{2, n\}$ and $\mathcal{X}_{n}=\{1, n-1\}$. For $2 \leq i \leq n$, let $\mathcal{X}_{i}=\{i-1, i+1\}$. Let $\mathcal{H}=\mathcal{H}(n, n, \mathcal{X}, f)$. Notice that the side information graph $\mathcal{G}$ is the odd cycle of length $n$. Therefore, $\alpha(\mathcal{H})=\alpha(\mathcal{G})=\ell$. From [4], $\kappa_{2}(\mathcal{H})=\operatorname{minrk}_{2}(\mathcal{G})=\ell+1$. From the $\alpha$-bound,

$$
\mathcal{N}_{2}[\delta, \mathcal{H}] \geq N_{2}[\ell, 2 \delta+1] .
$$

From the Singleton bound,

$$
\mathcal{N}_{2}[\delta, \mathcal{H}] \geq(\ell+1)+2 \delta
$$

As there are no nontrivial binary MDS codes, we have

$$
N_{2}[\ell, 2 \delta+1] \geq \ell+2 \delta+1
$$

for all $\delta>0$. Therefore, in this case the $\alpha$-bound is at least as good as the Singleton bound.

### 3.4 Random codes

In this section we prove an implicit upper bound on the optimal length of the ECIC's. The proof is based on constructing a random ECIC and analyzing its parameters.

Theorem 3.4.1. Let $\mathcal{H}=\mathcal{H}(m, n, \mathcal{X}, f)$ describe an ICSI instance. Then there exists a linear $(\delta, \mathcal{H})$-ECIC over $\mathbb{F}_{q}$ of length $N$ if

$$
\sum_{i \in[m]} q^{n-\left|\mathcal{X}_{i}\right|-1}<\frac{q^{N}}{V_{q}(N, 2 \delta)}
$$

where

$$
V_{q}(N, 2 \delta)=\sum_{\ell=0}^{2 \delta}\binom{N}{\ell}(q-1)^{\ell}
$$

is the volume of the $q$-ary sphere in $\mathbb{F}_{q}^{N}$.

Proof. We construct a random $n \times N$ matrix $\boldsymbol{L}$ over $\mathbb{F}_{q}$, row by row. Each row is selected independently of other rows, uniformly over $\mathbb{F}_{q}^{N}$. Define vector spaces

$$
\boldsymbol{M}_{i} \triangleq \operatorname{span}_{q}\left(\left\{\boldsymbol{L}_{j}: j \in \mathcal{Y}_{i}\right\}\right)
$$

for all $i \in[m]$. We also define the following events:

$$
\forall i \in[m]: \quad \text { Event } E_{i} \triangleq\left\{\mathrm{~d}\left(\boldsymbol{L}_{f(i)}, \boldsymbol{M}_{i}\right)<2 \delta+1\right\}
$$

and

$$
\text { Event } E_{F a i l} \triangleq\left\{\boldsymbol{L} \text { does not correspond to a }(\delta, \mathcal{H}) \text {-ECIC over } \mathbb{F}_{q}\right\}
$$

The event $E_{i}$ represents the situation when the receiver $R_{i}$ cannot recover $x_{f(i)}$. Then, by Corollary 3.1.4, the event $E_{\text {Fail }}$ is equivalent to $\cup_{i \in[m]} E_{i}$. Therefore,

$$
\begin{equation*}
\operatorname{Pr}\left(E_{\text {Fail }}\right)=\operatorname{Pr}\left(\bigcup_{i \in[m]} E_{i}\right) \leq \sum_{i \in[m]} \operatorname{Pr}\left(E_{i}\right) \tag{3.7}
\end{equation*}
$$

For a particular event $E_{i}, i \in[m]$,

$$
\begin{equation*}
\operatorname{Pr}\left(E_{i}\right) \leq \frac{q^{\left|\mathcal{Y}_{i}\right|} V_{q}(N, 2 \delta)}{q^{N}} \tag{3.8}
\end{equation*}
$$

There exists a matrix $\boldsymbol{L}$ corresponding to a $(\delta, \mathcal{H})$-ECIC over $\mathbb{F}_{q}$ if $\operatorname{Pr}\left(E_{\text {Fail }}\right)<1$. It is enough to require that the right-hand side of (3.7) is smaller than 1. By plugging in the expression in (3.8), we obtain a sufficient condition on the existence of a linear $(\delta, \mathcal{H})$-ECIC over $\mathbb{F}_{q}$ :

$$
\frac{V_{q}(N, 2 \delta)}{q^{N}} \sum_{i \in[m]} q^{\left|\mathcal{Y}_{i}\right|}<1 .
$$

Remark 3.4.2. The bound in Theorem 3.4.1 does not take into account the structure of the sets $\mathcal{X}_{i}$ 's, other than their cardinalities. Therefore, this bound generally is weaker than the $\kappa$-bound. On the other hand, for a particular ICSI instance, it is easier to calculate this bound, while computing the $\kappa$-bound in general is a hard problem.

Remark 3.4.3. The bound in Theorem 3.4.1 implies a bound on $\kappa_{q}(\mathcal{H})$, which is tight for some $\mathcal{H}$. Indeed, fix $\delta=0$. The bound implies that there exists a linear index code of length $N$ whenever

$$
\begin{equation*}
\sum_{i \in[m]} q^{n-\left|\mathcal{X}_{i}\right|-1}<q^{N} . \tag{3.9}
\end{equation*}
$$

Let $m=n=2 \ell+1(\ell \geq 2)$, and $f(i)=i$ for all $i \in[n]$. Let $\mathcal{X}_{1}=[n] \backslash\{1,2, n\}$ and $\mathcal{X}_{n}=[n] \backslash\{1, n-1, n\}$. For $2 \leq i \leq n-1$, let $\mathcal{X}_{i}=[n] \backslash\{i-1, i, i+1\}$. Let $\mathcal{H}=\mathcal{H}(n, n, \mathcal{X}, f)$ be the corresponding side information hypergraph. The side information graph $\mathcal{G}$ is the complement of the odd cycle of length $n$. We have $\left|\mathcal{X}_{i}\right|=2 \ell-2$ for all $i \in[n]$. Then (3.9) becomes

$$
N>2+\log _{q}(2 \ell+1)
$$

If $q>2 \ell+1$ then we obtain $N \geq 3$. Observe that in this case $\kappa_{q}(\mathcal{H})=\operatorname{minrk}_{q}(\mathcal{G})=3$ (see [2, Claim A.1]), and thus the bound is tight.

### 3.5 Strongly Secure Linear Error-Correcting Index Codes

In this section we consider an active adversary who not only owns some side information and eavesdrops some transmissions, but also corrupts some transmissions from $S$. Note that a simple concatenation of an error-correcting scheme and a security scheme may not necessarily work. Indeed, applying an error-correcting code on top of a secure index code may sabotage the security of the scheme. Conversely, if error-correcting index coding is followed by security coding, the scheme is no longer resistant to errors.

### 3.5.1 A Lower Bound on the Length of a Strongly Secure Linear Error-Correcting

 Index CodeThe notion of randomized index codes introduced in Subsection 2.4.1 can be naturally extended to that of $\delta$-error-correcting randomized index codes. Then we have the following definition.

Definition 3.5.1. A randomized index code is referred to as a $(\mu, t)$-strongly secure $\delta$-error-correcting randomized index code if it is not only $(\mu, t)$-strongly secure (see Definition 2.4.2) but also capable of correcting $\delta$ errors (see Definition 3.1.1).

The next theorem establishes a lower bound on the length of a $(\mu, t)$-strongly secure $\delta$-error-correcting linear randomized index code.

Theorem 3.5.2. The length of a $\mu, t)$-strongly secure $\delta$-error-correcting linear $\eta$ randomized $(m, n, \mathcal{X}, f)-I C$ over $\mathbb{F}_{q}$ is at least $\kappa_{q}+\mu+2 \delta$.

Proof. Let $\boldsymbol{L}$ correspond to a ( $\mu, t$ )-strongly secure $\delta$-error-correcting $\eta$-randomized $(m, n, \mathcal{X}, f)$-IC over $\mathbb{F}_{q}$. Let $\boldsymbol{L}^{\prime}$ be the matrix obtained from $\boldsymbol{L}$ by deleting any $2 \delta$ columns of $\boldsymbol{L}$. Since $\boldsymbol{L}$ corresponds to a $\delta$-error-correcting index code, by Lemma 3.1.3 it satisfies

$$
\mathrm{wt}\left(\sum_{i \in K} z_{i} \boldsymbol{L}_{i}\right) \geq 2 \delta+1
$$

for all $K \in \mathcal{J}(\mathcal{H})$ and all choices of nonzero $z_{i} \in \mathbb{F}_{q}, i \in K$. We deduce that the rows $\boldsymbol{L}_{i}^{\prime \prime}$ s of $\boldsymbol{L}^{\prime}$ also satisfy

$$
\mathrm{wt}\left(\sum_{i \in K} z_{i} \boldsymbol{L}_{i}^{\prime}\right) \geq 1
$$

By Corollary 3.1.5, $\boldsymbol{L}^{\prime}$ still corresponds to an $\eta$-randomized ( $m, n, \mathcal{X}, f$ )-IC over $\mathbb{F}_{q}$. Since less information about $\boldsymbol{x}$ is broadcast if $\boldsymbol{L}^{\prime}$ is used, we deduce that $\boldsymbol{L}^{\prime}$ also corresponds to a $(\mu, t)$-strongly secure $\eta$-randomized $(m, n, \mathcal{X}, f)$-IC over $\mathbb{F}_{q}$.

Therefore, by Theorem 3.2.5, $\boldsymbol{L}^{\prime}$ has at least $\kappa_{q}+\mu$ columns. We then deduce that $\boldsymbol{L}$ has at least $\kappa_{q}+\mu+2 \delta$ columns.

### 3.5.2 A Construction of an Optimal Strongly Secure Linear Error-Correcting Index Code

In this subsection we provide a construction that produces randomized index codes attaining the bound established in Theorem 3.5.2. This construction requires $q$ to be as large as $\kappa_{q}+\mu+2 \delta$.

Construction B: Let $\boldsymbol{L}^{(0)}$ correspond to a linear $(m, n, \mathcal{X}, f)$-IC over $\mathbb{F}_{q}$ of optimal length $\kappa_{q}$. Let $\boldsymbol{M}$ be a generator matrix of an $\left[N=\kappa_{q}+\mu+2 \delta, \kappa_{q}+\mu, 2 \delta+1\right]_{q^{-}}$ MDS code, so that the last $\mu$ rows of $\boldsymbol{M}$ form a generator matrix of another MDS code. For instance, we can take a generator matrix of a Generalized Reed-Solomon code (see, for instance [39])

$$
\boldsymbol{M}=\left(\begin{array}{cccc}
1 & 1 & \cdots & 1 \\
\alpha_{1} & \alpha_{2} & \cdots & \alpha_{N} \\
\vdots & \vdots & \cdots & \vdots \\
\alpha_{1}^{\kappa_{q}-1} & \alpha_{2}^{\kappa_{q}-1} & \cdots & \alpha_{N}^{\kappa_{q}-1} \\
\hline \alpha_{1}^{\kappa_{q}} & \alpha_{2}^{\kappa_{q}} & \cdots & \alpha_{N}^{\kappa_{q}} \\
\alpha_{1}^{\kappa_{q}+1} & \alpha_{2}^{\kappa_{q}+1} & \cdots & \alpha_{N}^{\kappa_{q}+1} \\
\vdots & \vdots & \cdots & \vdots \\
\alpha_{1}^{\kappa_{q}+\mu-1} & \alpha_{2}^{\kappa_{q}+\mu-1} & \cdots & \alpha_{N}^{\kappa_{q}+\mu-1}
\end{array}\right)
$$

where $\alpha_{1}, \alpha_{2}, \ldots, \alpha_{N}$ are pairwise distinct elements in $\mathbb{F}_{q}$. Let $\boldsymbol{P}$ be the submatrix of $\boldsymbol{M}$ formed by the first $\kappa_{q}$ rows, and $\boldsymbol{Q}$ the submatrix formed by the last $\mu$ rows
of $\boldsymbol{M}$. Let

$$
L=\left(\frac{\boldsymbol{L}^{(0)} \boldsymbol{P}}{Q}\right)
$$

Lemma 3.5.3. The matrix $\boldsymbol{L}$ in Construction $B$ corresponds to a $\delta$-error-correcting $\mu$-randomized $(m, n, \mathcal{X}, f)$-IC over $\mathbb{F}_{q}$.

Proof. Let $\boldsymbol{g} \in \mathbb{F}_{q}^{\mu}$ be a randomized vector, uniformly distributed over $\mathbb{F}_{q}^{\mu}$. The encoding process is as follows

$$
\boldsymbol{x} \mapsto(\boldsymbol{x} \mid \boldsymbol{g}) \boldsymbol{L}=\boldsymbol{x} \boldsymbol{L}^{(0)} \boldsymbol{P}+\boldsymbol{g} \boldsymbol{Q}=\left(\boldsymbol{x} \boldsymbol{L}^{(0)} \mid \boldsymbol{g}\right) \boldsymbol{M}
$$

Since $\boldsymbol{M}$ is a generator matrix of a $\delta$-error-correcting code, if at most $\delta$ errors occur, then each receiver is able to recover $\left(\boldsymbol{x} \boldsymbol{L}^{(0)} \mid \boldsymbol{g}\right)$. Therefore, for each $i \in[m]$, the receiver $R_{i}$ can recover $\boldsymbol{x} \boldsymbol{L}^{(0)}$, and hence, can also recover $x_{f(i)}$, as $\boldsymbol{L}^{(0)}$ corresponds to a linear $(m, n, \mathcal{X}, f)$-IC over $\mathbb{F}_{q}$.

Lemma 3.5.4. The matrix $\boldsymbol{L}$ in Construction $A$ corresponds to a $(\mu, t)$-strongly secure $\mu$-randomized $(m, n, \mathcal{X}, f)$-IC over $\mathbb{F}_{q}$.

Proof. The proof of this lemma is the same as that of Lemma 2.4.9.

From Theorem 3.5.2, Lemma 3.5.3, and Lemma 3.5.4, we have the following theorem.

Theorem 3.5.5. The length of an optimal ( $\mu, t$ )-strongly secure $\delta$-error-correcting linear $\eta$-randomized $(m, n, \mathcal{X}, f)$-IC over $\mathbb{F}_{q}$ is at least $\kappa_{q}+\mu+2 \delta$. Moreover, the code based on the matrix $\boldsymbol{L}$ established in Construction $B$ achieves this bound ( $q \geq$ $\left.\kappa_{q}+\mu+2 \delta\right)$.

### 3.6 Syndrome Decoding

Consider the $(\delta, \mathcal{H})$-ECIC over $\mathbb{F}_{q}$ based on a matrix $\boldsymbol{L}$. Suppose that the receiver $R_{i}, i \in[m]$, receives the vector

$$
\begin{equation*}
\boldsymbol{y}^{(i)}=\boldsymbol{x} \boldsymbol{L}+\boldsymbol{\epsilon}^{(i)}, \tag{3.10}
\end{equation*}
$$

where $\boldsymbol{x} \boldsymbol{L}$ is the codeword transmitted by $S$, and $\boldsymbol{\epsilon}^{(i)}$ is the error pattern affecting this codeword.

In the classical coding theory, the transmitted vector $\boldsymbol{c}$, the received vector $\boldsymbol{y}$, and the error pattern $\boldsymbol{e}$ are related by $\boldsymbol{y}=\boldsymbol{c}+\boldsymbol{e}$. Therefore, if $\boldsymbol{y}$ is known to the receiver, then there is a one-to-one correspondence between the values of unknown vectors $\boldsymbol{c}$ and $\boldsymbol{e}$. For index coding, however, this is no longer the case. The following lemma shows that, in order to recover the message $x_{f(i)}$ from $\boldsymbol{y}^{(i)}$ using (3.10), it is sufficient to find just one vector from a set of possible error patterns. This set is defined as follows:

$$
\mathcal{L}_{i}\left(\boldsymbol{\epsilon}^{(i)}\right)=\left\{\boldsymbol{\epsilon}^{(i)}+\boldsymbol{z}: \boldsymbol{z} \in \operatorname{span}_{q}\left(\left\{\boldsymbol{L}_{j}\right\}_{j \in \mathcal{Y}_{i}}\right)\right\} .
$$

We henceforth refer to the set $\mathcal{L}_{i}\left(\boldsymbol{\epsilon}^{(i)}\right)$ as the set of relevant error patterns.
Lemma 3.6.1. Assume that the receiver $R_{i}$ receives $\boldsymbol{y}^{(i)}$.

1. If $R_{i}$ knows the message $x_{f(i)}$ then it is able to determine the set $\mathcal{L}_{i}\left(\boldsymbol{\epsilon}^{(i)}\right)$.
2. If $R_{i}$ knows some vector $\hat{\boldsymbol{\epsilon}} \in \mathcal{L}_{i}\left(\boldsymbol{\epsilon}^{(i)}\right)$ then it is able to determine $x_{f(i)}$.
3. From (3.10), we have

$$
\begin{equation*}
\boldsymbol{y}^{(i)}=x_{f(i)} \boldsymbol{L}_{f(i)}+\boldsymbol{x}_{\mathcal{X}_{i}} \boldsymbol{L}_{\mathcal{X}_{i}}+\boldsymbol{x}_{\mathcal{Y}_{i}} \boldsymbol{L}_{\mathcal{Y}_{i}}+\boldsymbol{\epsilon}^{(i)} . \tag{3.11}
\end{equation*}
$$

If $R_{i}$ knows $x_{f(i)}$, then it is also able to determine

$$
\boldsymbol{\epsilon}^{(i)}+\boldsymbol{x}_{\mathcal{Y}_{i}} \boldsymbol{L}_{\mathcal{Y}_{i}}=\boldsymbol{y}^{(i)}-x_{f(i)} \boldsymbol{L}_{f(i)}-\boldsymbol{x}_{\mathcal{X}_{i}} \boldsymbol{L}_{\mathcal{X}_{i}} \in \mathcal{L}_{i}\left(\boldsymbol{\epsilon}^{(i)}\right)
$$

Since $R_{i}$ has knowledge of $\boldsymbol{L}$, it is also able to determine the whole $\mathcal{L}_{i}\left(\boldsymbol{\epsilon}^{(i)}\right)$.
2. Suppose that $R_{i}$ knows a vector

$$
\hat{\boldsymbol{\epsilon}}=\boldsymbol{\epsilon}^{(i)}+\sum_{j \in \mathcal{Y}_{i}} z_{j} \boldsymbol{L}_{j} \in \mathcal{L}_{i}\left(\boldsymbol{\epsilon}^{(i)}\right)
$$

for some $\boldsymbol{z} \in \mathbb{F}_{q}^{\left|\mathcal{Y}_{i}\right|}$. We show that $R_{i}$ is able then to determine $x_{f(i)}$. Indeed, we rewrite (3.11) as

$$
\begin{equation*}
\boldsymbol{y}^{(i)}=x_{f(i)} \boldsymbol{L}_{f(i)}+\boldsymbol{x}_{\mathcal{X}_{i}} \boldsymbol{L}_{\mathcal{X}_{i}}+\left(\boldsymbol{x}_{\mathcal{Y}_{i}}-\boldsymbol{z}\right) \boldsymbol{L}_{\mathcal{Y}_{i}}+\hat{\boldsymbol{\epsilon}} . \tag{3.12}
\end{equation*}
$$

The receiver $R_{i}$ can find some solution of the equation

$$
\begin{equation*}
\boldsymbol{y}^{(i)}=\hat{x}_{f(i)} \boldsymbol{L}_{f(i)}+\boldsymbol{x}_{\mathcal{X}_{i}} \boldsymbol{L}_{\mathcal{X}_{i}}+\hat{\boldsymbol{x}}_{\mathcal{Y}_{i}} \boldsymbol{L}_{\mathcal{Y}_{i}}+\hat{\boldsymbol{\epsilon}}, \tag{3.13}
\end{equation*}
$$

with respect to the unknowns $\hat{x}_{f(i)}$ and $\hat{\boldsymbol{x}}_{\mathcal{Y}_{i}}$. Observe that (3.13) has at least one solution due to (3.12).

From (3.12) and (3.13), we deduce that

$$
\mathbf{0}=\left(\hat{x}_{f(i)}-x_{f(i)}\right) \boldsymbol{L}_{f(i)}+\left(\hat{\boldsymbol{x}}_{\mathcal{Y}_{i}}-\boldsymbol{x}_{\mathcal{Y}_{i}}+\boldsymbol{z}\right) \boldsymbol{L}_{\mathcal{Y}_{i}} .
$$

This equality implies that $\hat{x}_{f(i)}=x_{f(i)}$ (otherwise, by Corollary 3.1.4, the sum in the right-hand side will have nonzero weight). Hence, $R_{i}$ is able to determine $x_{f(i)}$, as claimed.

We now describe a syndrome decoding algorithm for linear error-correcting index codes. From (3.11), we have

$$
\boldsymbol{y}^{(i)}-\boldsymbol{x}_{\mathcal{X}_{i}} \boldsymbol{L}_{\mathcal{X}_{i}}-\boldsymbol{\epsilon}^{(i)} \in \operatorname{span}_{q}\left(\left\{\boldsymbol{L}_{f(i)}\right\} \cup\left\{\boldsymbol{L}_{j}\right\}_{j \in \mathcal{Y}_{i}}\right) .
$$

Let $\mathscr{C}_{i}=\operatorname{span}_{q}\left(\left\{\boldsymbol{L}_{f(i)}\right\} \cup\left\{\boldsymbol{L}_{j}\right\}_{j \in \mathcal{Y}_{i}}\right)$, and let $\boldsymbol{H}^{(i)}$ be a parity check matrix of $\mathscr{C}_{i}$. We obtain that

$$
\begin{equation*}
\boldsymbol{H}^{(i)} \boldsymbol{\epsilon}^{(i)^{\mathrm{T}}}=\boldsymbol{H}^{(i)}\left(\boldsymbol{y}^{(i)}-\boldsymbol{x}_{\mathcal{X}_{i}} \boldsymbol{L}_{\mathcal{X}_{i}}\right)^{\mathrm{T}} \tag{3.14}
\end{equation*}
$$

Let $\boldsymbol{\beta}^{(i)}$ be a column vector defined by

$$
\begin{equation*}
\boldsymbol{\beta}^{(i)}=\boldsymbol{H}^{(i)}\left(\boldsymbol{y}^{(i)}-\boldsymbol{x}_{\mathcal{X}_{i}} \boldsymbol{L}_{\mathcal{X}_{i}}\right)^{\mathrm{T}} \tag{3.15}
\end{equation*}
$$

Observe that each $R_{i}$ is capable of determining $\boldsymbol{\beta}^{(i)}$. Then we can rewrite (3.14) as

$$
\boldsymbol{H}^{(i)} \boldsymbol{\epsilon}^{(i)^{\mathrm{T}}}=\boldsymbol{\beta}^{(i)}
$$

This leads us to the formulation of the following decoding procedure for $R_{i}$ (see Fig. 3.1).

Remark 3.6.2. Gaussian elimination can be used to solve (3.17) for $\hat{x}_{f(i)}$. However, since $\boldsymbol{L}$ also corresponds to an $(m, n, \mathcal{X}, f)$-IC over $\mathbb{F}_{q}$, there is more efficient way to do so. From Corollary 2.2.6, there exists a vector $\boldsymbol{u}^{(i)} \triangleleft \mathcal{X}_{i}$ satisfying $\boldsymbol{u}^{(i)}+\boldsymbol{e}_{f(i)} \in$

- Input: $\boldsymbol{y}^{(i)}, \boldsymbol{x}_{\mathcal{X}_{i}}, \boldsymbol{L}$.
- Step 1: Compute the syndrome

$$
\boldsymbol{\beta}^{(i)}=\boldsymbol{H}^{(i)}\left(\boldsymbol{y}^{(i)}-\boldsymbol{x}_{\mathcal{X}_{i}} \boldsymbol{L}_{\mathcal{X}_{i}}\right)^{\mathrm{T}}
$$

- Step 2: Find the lowest Hamming weight solution $\hat{\boldsymbol{\epsilon}}$ of the system

$$
\begin{equation*}
\boldsymbol{H}^{(i)} \hat{\boldsymbol{\epsilon}}^{\mathrm{T}}=\boldsymbol{\beta}^{(i)} \tag{3.16}
\end{equation*}
$$

- Step 3: Given that $\hat{\boldsymbol{x}}_{\mathcal{X}_{i}}=\boldsymbol{x}_{\mathcal{X}_{i}}$, solve the system for $\hat{x}_{f(i)}$ :

$$
\begin{equation*}
\boldsymbol{y}^{(i)}=\hat{\boldsymbol{x}} \boldsymbol{L}+\hat{\boldsymbol{\epsilon}} . \tag{3.17}
\end{equation*}
$$

- Output: $\hat{x}_{f(i)}$.

Fig. 3.1: Syndrome decoding procedure.
$\mathcal{C}(\boldsymbol{L})$. Hence $\boldsymbol{u}^{(i)}+\boldsymbol{e}_{f(i)}=\boldsymbol{\alpha} \boldsymbol{L}^{\mathrm{T}}$ for some $\boldsymbol{\alpha} \in \mathbb{F}_{q}^{N}$. Therefore

$$
\hat{x}_{f(i)}=\hat{\boldsymbol{x}}\left(\boldsymbol{u}^{(i)}+\boldsymbol{e}_{f(i)}\right)^{\mathrm{T}}-\hat{\boldsymbol{x}} \boldsymbol{u}^{(i)^{\mathrm{T}}}=\hat{\boldsymbol{x}} \boldsymbol{L} \boldsymbol{\alpha}^{\mathrm{T}}-\hat{\boldsymbol{x}} \boldsymbol{u}^{(i)^{\mathrm{T}}}=\left(\boldsymbol{y}^{(i)}-\hat{\boldsymbol{\epsilon}}\right) \boldsymbol{\alpha}^{\mathrm{T}}-\hat{\boldsymbol{x}} \boldsymbol{u}^{(i)^{\mathrm{T}}}
$$

With the knowledge of $\boldsymbol{L}$ and $\boldsymbol{x}_{\mathcal{X}_{i}}$, the receiver $R_{i}$ can determine $\boldsymbol{\alpha}$ and $\hat{\boldsymbol{x}} \boldsymbol{u}^{(i)^{\mathrm{T}}}$. Therefore, it can also determine $\hat{x}_{f(i)}$. Note that (3.17) may have more than one solution $\hat{\boldsymbol{x}}$ with $\hat{\boldsymbol{x}}_{\mathcal{X}_{i}}=\boldsymbol{x}_{\mathcal{X}_{i}}$. However, as shown in the next theorem, if at most $\delta$ errors occur in $\boldsymbol{y}^{(i)}$, then it always holds that $\hat{x}_{f(i)}=x_{f(i)}$.

Theorem 3.6.3. Let $\boldsymbol{y}^{(i)}=\boldsymbol{x} \boldsymbol{L}+\boldsymbol{\epsilon}^{(i)}$ be the vector received by $R_{i}$, and let $\mathrm{wt}\left(\boldsymbol{\epsilon}^{(i)}\right) \leq$ $\delta$. Assume that the procedure in Fig. 3.1 is applied to $\left(\boldsymbol{y}^{(i)}, \boldsymbol{x}_{\mathcal{X}_{i}}, \boldsymbol{L}\right)$. Then, its output satisfies $\hat{x}_{f(i)}=x_{f(i)}$.

Proof. By Lemma 3.6.1, it is sufficient to prove that $\hat{\boldsymbol{\epsilon}} \in \mathcal{L}_{i}\left(\boldsymbol{\epsilon}^{(i)}\right)$. Indeed, since

$$
\boldsymbol{H}^{(i)} \boldsymbol{\epsilon}^{(i)^{\mathrm{T}}}=\boldsymbol{H}^{(i)} \hat{\boldsymbol{\epsilon}}^{\mathrm{T}}=\boldsymbol{\beta}^{(i)},
$$

we have

$$
\boldsymbol{H}^{(i)}\left(\hat{\boldsymbol{\epsilon}}-\boldsymbol{\epsilon}^{(i)}\right)^{\mathrm{T}}=\mathbf{0}
$$

Hence, $\hat{\boldsymbol{\epsilon}}-\boldsymbol{\epsilon}^{(i)} \in \mathscr{C}_{i}$, and therefore,

$$
\begin{equation*}
\hat{\boldsymbol{\epsilon}}-\boldsymbol{\epsilon}^{(i)}=z_{f(i)} \boldsymbol{L}_{f(i)}+\sum_{j \in \mathcal{Y}_{i}} z_{j} \boldsymbol{L}_{j}, \tag{3.18}
\end{equation*}
$$

for some $z_{f(i)} \in \mathbb{F}_{q}$ and $z_{j} \in \mathbb{F}_{q}, j \in \mathcal{Y}_{i}$.
Since $\boldsymbol{\epsilon}^{(i)}$ is a solution of (3.16), and $\mathrm{wt}\left(\boldsymbol{\epsilon}^{(i)}\right) \leq \delta$, we deduce that $\mathrm{wt}(\hat{\boldsymbol{\epsilon}}) \leq \delta$ as well. Hence,

$$
\mathrm{wt}\left(z_{f(i)} \boldsymbol{L}_{f(i)}+\sum_{j \in \mathcal{Y}_{i}} z_{j} \boldsymbol{L}_{j}\right)=\mathrm{wt}\left(\hat{\boldsymbol{\epsilon}}-\boldsymbol{\epsilon}^{(i)}\right) \leq 2 \delta .
$$

Therefore, by Corollary 3.1.4, $z_{f(i)}=0$. Hence, $\hat{\boldsymbol{\epsilon}} \in \mathcal{L}_{i}\left(\boldsymbol{\epsilon}^{(i)}\right)$, as desired, and therefore $\hat{x}_{f(i)}=x_{f(i)}$.

Remark 3.6.4. Step 2 in Fig. 3.1 is computationally hard. Indeed, the problem of finding $\hat{\boldsymbol{\epsilon}}$ over $\mathbb{F}_{2}$ of the lowest weight satisfying

$$
\begin{equation*}
\boldsymbol{H}^{(i)} \hat{\boldsymbol{\epsilon}}^{\mathrm{T}}=\boldsymbol{\beta}^{(i)} \tag{3.19}
\end{equation*}
$$

for a given binary vector $\boldsymbol{\beta}^{(i)}$ corresponds to the decision problem COSET WEIGHTS, which was shown by Berlekamp et al. [7] to be NP-complete.

### 3.7 Static Error-Correcting Index Codes

### 3.7.1 Static Error-Correcting Index Codes

In previous sections of this chapter, we focus on designing an optimal linear $\delta$-errorcorrecting index code for a particular ICSI instance. When the parameters $m, n, \mathcal{X}$, and $f$ are changed, it is very likely that an error-correcting index code for the current instance can no longer be used for the new instance. Therefore, an interesting task is to design an error-correcting index code that works for a family of ICSI instances.

Definition 3.7.1. Let $\Gamma$ be a set of instances $(m, n, \mathcal{X}, f)$ for the ICSI problem. A $\delta$-error-correcting index code over $\mathbb{F}_{q}$ is said to be static under the set $\Gamma$ if it is a $\delta$-error-correcting $(m, n, \mathcal{X}, f)$-IC over $\mathbb{F}_{q}$ for all instances $(m, n, \mathcal{X}, f) \in \Gamma$.

Recall that an instance $(m, n, \mathcal{X}, f)$ can be described by the side information hypergraph $\mathcal{H}(m, n, \mathcal{X}, f)$. For a set $\Gamma$ of instances $(m, n, \mathcal{X}, f)$, let

$$
\begin{equation*}
\mathfrak{J}(\Gamma) \triangleq \bigcup_{(m, n, \mathcal{X}, f) \in \Gamma} \mathcal{J}(\mathcal{H}(m, n, \mathcal{X}, f)) \tag{3.20}
\end{equation*}
$$

where $\mathcal{J}(\mathcal{H}(m, n, \mathcal{X}, f))$ is defined as in (3.1). We also define

$$
n(\Gamma) \triangleq \max \{n: \quad(m, n, \mathcal{X}, f) \in \Gamma\}
$$

Lemma 3.7.2. The $n(\Gamma) \times N$ matrix $\boldsymbol{L}$ corresponds to a $\delta$-error-correcting index code that is static under $\Gamma$ if and only if

$$
\mathrm{wt}\left(\sum_{i \in K} z_{i} \boldsymbol{L}_{i}\right) \geq 2 \delta+1
$$

for all $K \in \mathfrak{J}(\Gamma)$ and for all choices of $z_{i} \in \mathbb{F}_{q}^{*}, i \in K$.
Proof. The proof is immediate from Definition 3.7.1 and Lemma 3.1.3. Note that when $\boldsymbol{L}$ is used for an instance $(m, n, \mathcal{X}, f) \in \Gamma$ with $n<n(\Gamma)$, then the last $n(\Gamma)-n$ rows of $\boldsymbol{L}$ are simply discarded.

One particular family of interest is $\Gamma(n, \rho)$, the family that contains all instances where each receiver owns at least $n-\rho$ messages as its side information. More formally,

$$
\Gamma(n, \rho)=\left\{\left(m, n^{\prime}, \mathcal{X}, f\right): n^{\prime} \leq n \text { and } \forall i \in[m],\left|\mathcal{X}_{i}\right| \geq n-\rho\right\}
$$

A $\delta$-error-correcting index code that is static under $\Gamma(n, \rho)$ will provide successful communication between the sender and the receivers under the presence of at most $\delta$ errors, despite a possible change of the sets of side information, a change of the set of receivers, and a change of the demand function, as long as each receiver still possesses at least $n-\rho$ messages.

In the rest of this section, we assume that $N \geq 1, n \geq \rho \geq 1$ and $\delta \geq 0$.

Definition 3.7.3. An $n \times N$ matrix $\boldsymbol{L}$ is said to satisfy the $(\rho, \delta)$-Property if any nontrivial linear combination of at most $\rho$ rows of $\boldsymbol{L}$ has weight at least $2 \delta+1$.

Proposition 3.7.4. The $n \times N$ matrix $\boldsymbol{L}$ corresponds to a $\delta$-error-correcting linear index code that is static under $\Gamma(n, \rho)$, if and only if $\boldsymbol{L}$ satisfies the $(\rho, \delta)$-Property.

Proof. By Lemma 3.7.2, it suffices to show that $\mathfrak{J}(\Gamma(n, \rho))$ is the collection of all nonempty subsets of $[n]$, whose cardinalities are not greater than $\rho$.

Consider an instance $\left(m, n^{\prime}, \mathcal{X}, f\right) \in \Gamma(n, \rho)$. For all $i \in[m]$, we have $\left|\mathcal{X}_{i}\right| \geq n-\rho$
and $\mathcal{Y}_{i}=\left[n^{\prime}\right] \backslash\left(f(i) \cup \mathcal{X}_{i}\right)$, and thus we deduce that

$$
\left|\mathcal{Y}_{i}\right| \leq n^{\prime}-1-(n-\rho) \leq n^{\prime}-1-\left(n^{\prime}-\rho\right)=\rho-1 .
$$

Hence by (3.1), the cardinality of each set in $\mathcal{J}\left(\mathcal{H}\left(m, n^{\prime}, \mathcal{X}, f\right)\right)$ is at most

$$
1+(\rho-1)=\rho
$$

Therefore, due to (3.20), every set in $\mathfrak{J}(\Gamma(n, \rho))$ has at most $\rho$ elements.
It remains to show that every nonempty subset of $[n]$ whose cardinality is at most $\rho$ belongs to $\mathfrak{J}(\Gamma(n, \rho))$. Consider an arbitrary $\rho^{\prime}$-subset $K=\left\{i_{1}, i_{2}, \ldots, i_{\rho^{\prime}}\right\}$ of $[n]$, with $1 \leq \rho^{\prime} \leq \rho$. Consider an instance $(m=1, n, \mathcal{X}, f) \in \Gamma(n, \rho)$ with $\mathcal{X}_{1}=[n] \backslash K$ and $f(1)=i_{1}$. Since

$$
\mathcal{Y}_{1}=K \backslash\left\{i_{1}\right\}
$$

we have

$$
K=\left\{i_{1}\right\} \cup \mathcal{Y}_{1} \in \mathcal{J}(\mathcal{H}(m, n, \mathcal{X}, f)) \subseteq \mathfrak{J}(\Gamma(n, \rho))
$$

The proof follows.

### 3.7.2 An Application: Weakly Resilient Functions

In this section we introduce the notion of weakly resilient functions. Hereafter, we restrict the discussion to the binary alphabet.

The concept of binary resilient functions was first introduced by Chor et al. [17] and independently by Bennet et al. [6].

Definition 3.7.5. A function $\boldsymbol{f}: \mathbb{F}_{2}^{N} \rightarrow \mathbb{F}_{2}^{n}$ is said to be $t$-resilient if $\boldsymbol{f}$ satisfies the property that it runs through every possible output $n$-tuple an equal number of
times when $t$ arbitrary inputs are fixed and the remaining $N-t$ inputs run through all the $2^{N-t}$ input tuples once. Moreover, if $\boldsymbol{f}$ is a linear transformation then it is called a linear $t$-resilient function. We also refer to $t$ as the resiliency of $\boldsymbol{f}$.

The applications of resilient functions can be found in fault-tolerant distributed computing, quantum cryptographic key distribution [17], privacy amplification [6] and random sequence generation for stream ciphers [13]. The equivalence between a linear error-correcting code and a resilient function was established by Chor [17].

Theorem 3.7.6 ([17]). Let $\boldsymbol{L}$ be an $n \times N$ binary matrix. Then $\boldsymbol{L}$ is a generator matrix of a linear error-correcting code with minimum distance $d=t+1$ if and only if $\boldsymbol{f}(\boldsymbol{z})=\boldsymbol{z} \boldsymbol{L}^{T}$ is $t$-resilient.

Remark 3.7.7. Vectorial boolean functions with certain properties are useful for design of stream ciphers. These properties include high resiliency and high nonlinearity (see, for instance, [13]). Nevertheless, linear resilient functions are still particularly interesting, since they can be transformed into highly nonlinear resilient functions with the same parameters. This can be achieved by a composition of the linear function with a highly nonlinear permutation (see $[33,70]$ for more details).

Below we introduce a definition of a $\rho$-weakly $t$-resilient function, which is a weaker version of a $t$-resilient function.

Definition 3.7.8. A function $\boldsymbol{f}: \mathbb{F}_{2}^{N} \rightarrow \mathbb{F}_{2}^{n}$ is said to be $\rho$-weakly $t$-resilient if $\boldsymbol{f}$ satisfies the property that every set of $\rho$ coordinates of $\boldsymbol{f}$ runs through every possible output $\rho$-tuple an equal number of times when $t$ arbitrary inputs are fixed and the remaining $N-t$ inputs run through all the $2^{N-t}$ input tuples once.

Remark 3.7.9. A $\rho$-weakly $t$-resilient function $\boldsymbol{f}: \mathbb{F}_{2}^{N} \rightarrow \mathbb{F}_{2}^{n}$ can be viewed as a collection of $\binom{n}{\rho}$ different $t$-resilient functions $\mathbb{F}_{2}^{N} \rightarrow \mathbb{F}_{2}^{\rho}$, each such function is obtained by taking some $\rho$ coordinates in the image of $\boldsymbol{f}$. Similarly to [17], consider a scenario, in which two parties are sharing a secret key, which consists of $N$ randomly selected bits. Suppose that at some moment $t$ out of the $N$ bits of the key are leaked to an adversary. By applying a $t$-resilient function to the current $N$-bit key, two parties are able to obtain a completely new and secret key of $n$ bits, without requiring any communication or randomness generation. However, if the parties use various parts of the key for various purposes, they may require several secret $\rho$-bit keys rather than one secret $n$-bit key. In that case a $\rho$-weakly $t$-resilient function can be used. By applying a $\rho$-weakly $t$-resilient function to the current $N$-bit key, the parties obtain a set of $\binom{n}{\rho}$ different $\rho$-bit keys, each key is new and secret (however these keys might not be independent of each other). With the same number of inputs $N$ and outputs $n$, a weakly resilient function may have strictly higher resiliency than a resilient function (see Remark 3.7.17).

Theorem 3.7.10. Let $\boldsymbol{L}$ be an $n \times N$ binary matrix. Then $\boldsymbol{L}$ satisfies the $(\rho, \delta)$ Property if and only if the function $\boldsymbol{f}: \mathbb{F}_{2}^{N} \rightarrow \mathbb{F}_{2}^{n}$ defined by $\boldsymbol{f}(\boldsymbol{z})=\boldsymbol{z} \boldsymbol{L}^{T}$ is $\rho$-weakly $2 \delta$-resilient.

Proof.

1. Suppose that $\boldsymbol{L}$ satisfies the $(\rho, \delta)$-Property. Take any $\rho$-subset $K \subseteq[n]$. By Definition 3.7.3, the $\rho \times N$ submatrix $\boldsymbol{L}_{K}$ of $\boldsymbol{L}$ is a generating matrix of the error-correcting code with the minimum distance $\geq 2 \delta+1$. By Theorem 3.7.6, the function $\boldsymbol{f}_{K}: \mathbb{F}_{2}^{N} \rightarrow \mathbb{F}_{2}^{\rho}$ defined by $\boldsymbol{f}_{K}(\boldsymbol{z})=\boldsymbol{z} \boldsymbol{L}_{K}^{\mathrm{T}}$ is $2 \delta$-resilient. Since $K$ is an arbitrary $\rho$-subset of $[n]$, the function $\boldsymbol{f}$ is $\rho$-weakly $2 \delta$-resilient.
2. Conversely, assume that the function $\boldsymbol{f}$ is $\rho$-weakly $2 \delta$-resilient. Take any subset $K \subseteq[n]$ where $|K|=\rho$. Then the function $\boldsymbol{f}_{K}: \mathbb{F}_{2}^{N} \rightarrow \mathbb{F}_{2}^{\rho}$ defined by $\boldsymbol{f}_{K}(\boldsymbol{z})=\boldsymbol{z} \boldsymbol{L}_{K}^{\mathrm{T}}$ is $2 \delta$-resilient. Therefore, by Theorem 3.7.6, $\boldsymbol{L}_{K}$ is a generating matrix of a linear code with minimum distance $2 \delta+1$. Since $K$ is an arbitrary $\rho$-subset of $[n]$, by Proposition 3.7.4 $\boldsymbol{L}$ satisfies the $(\rho, \delta)$-Property.

### 3.7.3 Bounds and Constructions

In this section we study the problem of constructing a matrix $\boldsymbol{L}$ satisfying the $(\rho, \delta)$ Property. Such an $\boldsymbol{L}$ with the minimum number of columns is called optimal. It turns out that this is a special case of the problem of finding an optimal linear error-correcting index code. However, in this special case, as more structure is given to the sets of side information, the bounds and constructions based on $\alpha(\mathcal{H})$ and $\kappa_{q}(\mathcal{H})$ (see Section 3.2) are much simpler.

Recall that in the proof of Proposition 3.7.4 it is shown that

$$
\mathfrak{J}(\Gamma(n, \rho))=\bigcup_{i=1}^{\rho}\binom{[n]}{i}
$$

the set of all nonempty subsets of $[n]$ of cardinality at most $\rho$. Consider an instance $\left(m^{*}, n, \mathcal{X}^{*}, f^{*}\right)$ satisfying

$$
\begin{equation*}
\mathcal{J}\left(\mathcal{H}^{*}\right)=\mathfrak{J}(\Gamma(n, \rho))=\bigcup_{i=1}^{\rho}\binom{[n]}{i} \tag{3.21}
\end{equation*}
$$

where $\mathcal{H}^{*}=\mathcal{H}\left(m^{*}, n, \mathcal{X}^{*}, f^{*}\right)$ is the side information hypergraph corresponding to this instance. Such an instance can be constructed as follows. For each subset $K=$ $\left\{i_{1}, i_{2}, \ldots, i_{\rho^{\prime}}\right\} \subseteq[n]\left(1 \leq \rho^{\prime} \leq \rho\right)$, we introduce a receiver which requests the message $x_{i_{1}}$, and has side information $\left\{x_{j}: j \in[n] \backslash K\right\}$. It is straightforward to verify that we
indeed obtain an instance $\left(m^{*}, n, \mathcal{X}^{*}, f^{*}\right)$ satisfying (3.21). The problem of designing an optimal matrix $\boldsymbol{L}$ satisfying the $(\rho, \delta)$-Property then becomes equivalent to the problem of finding an optimal $\left(\delta, \mathcal{H}^{*}\right)$-ECIC. Thus, $\mathcal{N}_{q}\left[\mathcal{H}^{*}, \delta\right]$ is equal to the number of columns in an optimal matrix that satisfies the $(\rho, \delta)$-Property. In other words, $\mathcal{N}_{q}\left[\mathcal{H}^{*}, \delta\right]$ is the length of an optimal $\delta$-error-correcting linear index code that is static under $\Gamma(n, \rho)$.

The corresponding $\alpha$-bound and $\kappa$-bound for $\mathcal{N}_{q}\left[\mathcal{H}^{*}, \delta\right]$ can be stated as follows.

Theorem 3.7.11. Let $\rho^{*}$ be the smallest number such that an $\left[n, n-\rho^{*}, \geq \rho+1\right]_{q}$ code exists. Then we have

$$
N_{q}[\rho, 2 \delta+1] \leq \mathcal{N}_{q}\left[\delta, \mathcal{H}^{*}\right] \leq N_{q}\left[\rho^{*}, 2 \delta+1\right] .
$$

Proof. The first inequality follows from the $\alpha$-bound and from the fact that $\alpha\left(\mathcal{H}^{*}\right)=$ $\rho$, which is due to (3.21).

For the second inequality, it suffices to show that $\kappa_{q}\left(\mathcal{H}^{*}\right)=\rho^{*}$. By Corollary 3.1.5, a matrix $\boldsymbol{L}$ corresponds to an $\mathcal{H}^{*}$-IC if and only if $\left\{\boldsymbol{L}_{i}: i \in K\right\}$ is linear independent for every $K \in \mathcal{J}\left(\mathcal{H}^{*}\right)$. Since $\mathcal{J}\left(\mathcal{H}^{*}\right)$ is the set of all nonempty subsets of cardinality at most $\rho$, this is equivalent to saying that every set of at most $\rho$ rows of $\boldsymbol{L}$ is linear independent. This condition is equivalent to the condition that $\boldsymbol{L}^{\mathrm{T}}$ is a parity check matrix of an $[n, n-N, \geq \rho+1]_{q}$ code [49, Chapter 1]. Therefore, a linear $\mathcal{H}^{*}$-IC of length $N$ exists if and only if an $[n, n-N, \geq \rho+1]_{q}$ code exists. Since $\rho^{*}$ is the smallest number such that an $\left[n, n-\rho^{*}, \geq \rho+1\right]_{q}$ code exists, we conclude that $\kappa_{q}\left(\mathcal{H}^{*}\right)=\rho^{*}$.

Corollary 3.7.12. The length of an optimal $\delta$-error-correcting linear index code
over $\mathbb{F}_{q}$ that is static under $\Gamma(n, \rho)$ satisfies

$$
\mathcal{N}_{q}\left[\delta, \mathcal{H}^{*}\right] \geq \rho^{*}+2 \delta
$$

where $\rho^{*}$ is the smallest number such that an $\left[n, n-\rho^{*}, \geq \rho+1\right]_{q}$ code exists.

Proof. This is a straightforward corollary of Theorem 3.3.1 (the Singleton bound) and Theorem 3.7.11.

Remark 3.7.13. We can see that the three bounds on the length of an optimal linear error-correcting index code that is static under $\Gamma(n, \rho)$ established above can be computed explicitly by solely using the results from classical error-correcting codes. By contrast, the original bounds for general ECIC involve the computation of $\alpha(\mathcal{H})$ and $\kappa_{q}(\mathcal{H})$, which are hard problems.

Corollary 3.7.14. For $q \geq \max \{n-1, \rho+2 \delta-1\}$, the length of an optimal $\delta$-errorcorrecting linear index code over $\mathbb{F}_{q}$ that is static under $\Gamma(n, \rho)$ is $\rho+2 \delta$.

Proof. When $q \geq n-1$, an (optimal) $[n, n-\rho, \rho+1]_{q}$ code exists (one may choose an extended Reed-Solomon code [49, Chapter 11]). Therefore $\rho^{*}=\rho$. Following the lines of the proof of Theorem 3.7.11, there exists a $\delta$-error-correcting index code of length $N_{q}[\rho, 2 \delta+1]$, which is static under $\Gamma(n, \rho)$. As $q \geq \rho+2 \delta-1$, we have

$$
N_{q}[\rho, 2 \delta+1]=\rho+2 \delta
$$

(for example, by taking an extended RS code). Due to Corollary 3.7.12, this static error-correcting index code is optimal.

Remark 3.7.15. We observe from the proof of Theorem 3.7.11 that the problem of constructing an optimal linear (non-error-correcting) index code which is static under $\Gamma(n, \rho)$, is, in fact, equivalent to the problem of constructing a parity check matrix of a classical linear error-correcting code.

Example 3.7.16. Let $n=20, \rho=10, \delta=1$ and $q=2$. From [31], an optimal binary linear code of length 20 and minimum distance 11 has dimension 3 . We deduce that $\rho^{*}=17$. We also have $N_{2}[17,3]=22$. Theorem 3.7.11 implies the existence of a one-error-correcting index code of length 22 which can be used for any instance in which each receiver owns at least 10 out of (at most) 20 messages as side information. It also implies that the length of any such static error-correcting index code is at least $N_{2}[10,3]=14$. Corollary 3.7.12 provides a better lower bound, which is $19=17+2$.

Remark 3.7.17. Below we show that with the same number of inputs $N$ and outputs $n$, a weakly resilient function may have strictly higher resiliency $t$. From Example 3.7.16, there exists a linear vectorial Boolean function $\boldsymbol{f}:\left(\mathbb{F}_{2}\right)^{22} \rightarrow\left(\mathbb{F}_{2}\right)^{20}$ which is 10 -weakly 2 -resilient. According to [31], an optimal linear [22, 20] ${ }_{2}$ code has minimum distance $d=2$. Hence, due to Theorem 3.7.6, the resiliency of any linear vectorial Boolean function $\boldsymbol{g}:\left(\mathbb{F}_{2}\right)^{22} \rightarrow\left(\mathbb{F}_{2}\right)^{20}$ cannot exceed one.

The problem of constructing an $n \times N$ matrix $\boldsymbol{L}$ that satisfies the $(\rho, \delta)$-Property is a natural generalization of the problem of constructing the parity check matrix $\boldsymbol{H}$ of a linear $[n, k, d \geq \rho+1]_{q}$ code. Indeed, $\boldsymbol{H}$ is a parity check matrix of an $[n, k, d \geq$ $\rho+1]_{q}$ code if and only if every set of $\rho$ columns of $\boldsymbol{H}$ is linearly independent. Equivalently, any nontrivial linear combination of at most $\rho$ columns of $\boldsymbol{H}$ has weight at least one. For comparison, $\boldsymbol{L}$ satisfies the $(\rho, \delta)$-Property if and only if
any nontrivial linear combination of at most $\rho$ columns of $\boldsymbol{L}^{\mathrm{T}}$ has weight at least $2 \delta+1$.

Some classical methods for deriving bounds on the parameters of error-correcting codes can be generalized to the case of linear static error-correcting index codes. Below we present a Gilbert-Varshamov-like bound.

Theorem 3.7.18. If

$$
\sum_{i=1}^{\rho-1}\binom{n-1}{i}(q-1)^{i} \sum_{j=0}^{2 \delta}\binom{N}{j}(q-1)^{j}<q^{N}
$$

then there exists an $n \times N$ matrix $\boldsymbol{L}$ that satisfies the $(\rho, \delta)$-Property.

Proof. We build up the set $\mathcal{R}$ of rows of $\boldsymbol{L}$ one by one. The first row can be any vector in $\mathbb{F}_{q}^{N}$ of weight at least $2 \delta+1$. Now suppose we have chosen $r$ rows so that no nontrivial linear combination of at most $\rho$ among these $r$ rows have weight less than $2 \delta+1$. Recall that

$$
V_{q}(N, 2 \delta)=\sum_{\ell=0}^{2 \delta}\binom{N}{\ell}(q-1)^{\ell}
$$

denotes the volume of the $q$-ary sphere in $\mathbb{F}_{q}^{N}$. There are at most

$$
V_{q}(N, 2 \delta) \sum_{i=0}^{\rho-1}\binom{r}{i}(q-1)^{i}
$$

vectors that are at distance less than $2 \delta+1$ from any linear combination of at most $\rho-1$ among $r$ chosen rows (this includes vectors at distance less than $2 \delta+1$ from $\mathbf{0}$ ). If this quantity is smaller than $q^{N}$, then we can add another row to the set $\mathcal{R}$ so that no nontrivial linear combination of at most $\rho$ rows in $\mathcal{R}$ has weight less than $2 \delta+1$. We repeat this process until $r=n$.

Remark 3.7.19. If we apply Theorem 3.4.1 to the instance ( $m^{*}, n, \mathcal{X}^{*}, f^{*}$ ) defined in the beginning of this section, then we obtain a bound, which is somewhat weaker then its counterpart in Theorem 3.7.18, namely the $n \times N$ matrix $\boldsymbol{L}$ as above exists if

$$
\sum_{i=1}^{\rho}\binom{n}{i} q^{i-1}<\frac{q^{N}}{V_{q}(N, 2 \delta)}
$$

### 3.8 Nonlinear Error-Correcting Index Codes

We now turn our attention to nonlinear ECIC's. Analogous results for bounds on the length of optimal nonlinear ECIC's can be established.

Definition 3.8.1. Let $\mathfrak{E}$ be a nonlinear ECIC over $\mathbb{F}_{q}$. The size of $\mathfrak{E}$ is defined to be

$$
\operatorname{size}(\mathfrak{E})=\left|\left\{\mathfrak{E}(\boldsymbol{x}): \boldsymbol{x} \in \mathbb{F}_{q}^{n}\right\}\right|
$$

Definition 3.8.2. Let $(m, n, \mathcal{X}, f)$ be an ICSI instance and $\mathcal{H}=\mathcal{H}(m, n, \mathcal{X}, f)$.

1. The two vectors $\boldsymbol{x}$ and $\boldsymbol{x}^{\prime}$ in $\mathbb{F}_{q}^{n}$ are called confusable if there exists an $i \in[n]$ such that $x_{f(i)} \neq x_{f(i)}^{\prime}$ and yet $x_{j}=x_{j}^{\prime}$ for all $j \in \mathcal{X}_{i}$.
2. The confusion graph of $\mathcal{H}=\mathcal{H}(m, n, \mathcal{X}, f)$, denoted $\mathfrak{C}(\mathcal{H})$, is the graph on the vertex set $\mathbb{F}_{q}^{n}$, where two vertices $\boldsymbol{x}$ and $\boldsymbol{x}^{\prime}$ are adjacent if and only if they are confusable.

Lemma 3.8.3. Let $(m, n, \mathcal{X}, f)$ be an ICSI instance described by the side information hypergraph $\mathcal{H}$. Then $\mathfrak{E}: \mathbb{F}_{q}^{n} \rightarrow \mathbb{F}_{q}^{N}$ is a $(\delta, \mathcal{H})$-ECIC over $\mathbb{F}_{q}$ if and only if $\mathrm{d}\left(\mathfrak{E}(\boldsymbol{x}), \mathfrak{E}\left(\boldsymbol{x}^{\prime}\right)\right) \geq 2 \delta+1$ whenever $\boldsymbol{x}$ and $\boldsymbol{x}^{\prime}$ are adjacent in $\mathfrak{C}(\mathcal{H})$.

Proof. Suppose that $\mathrm{d}\left(\mathfrak{E}(\boldsymbol{x}), \mathfrak{E}\left(\boldsymbol{x}^{\prime}\right)\right) \geq 2 \delta+1$ for every pair of adjacent vertices $\boldsymbol{x}$ and $\boldsymbol{x}^{\prime}$ of $\mathfrak{C}(\mathcal{H})$. In other words, suppose that confusable vectors are encoded to
codewords of distance at least $2 \delta+1$ from each other. Then a nearest neighbor decoding will always be able to recover the transmitted vector of messages if the number of erroneous symbols in the received vector is at most $\delta$. Hence $\mathfrak{E}$ is a $(\delta, \mathcal{H})$-ECIC over $\mathbb{F}_{q}$.

Conversely, suppose that $\mathfrak{E}$ is a $(\delta, \mathcal{H})$-ECIC over $\mathbb{F}_{q}$. Consider a pair of confusable vectors $\boldsymbol{x}$ and $\boldsymbol{x}^{\prime}$. If $\mathrm{d}\left(\mathfrak{E}(\boldsymbol{x}), \mathfrak{E}\left(\boldsymbol{x}^{\prime}\right)\right) \leq 2 \delta$, there exists a vector $\boldsymbol{y} \in \mathbb{F}_{q}^{N}$ such that

$$
\mathrm{d}(\mathfrak{E}(\boldsymbol{x}), \boldsymbol{y}) \leq \delta,
$$

and

$$
\mathrm{d}\left(\mathfrak{E}\left(\boldsymbol{x}^{\prime}\right), \boldsymbol{y}\right) \leq \delta,
$$

If $\boldsymbol{y}$ is received, since $\boldsymbol{x}_{\mathcal{X}_{i}}=\boldsymbol{x}_{\mathcal{X}_{i}}^{\prime}$ and $x_{f(i)} \neq x_{f(i)}^{\prime}$, it is impossible for the receiver to determine its desired message. This contradicts our assumption that $\mathfrak{E}$ is a $(\delta, \mathcal{H})$ ECIC over $\mathbb{F}_{q}$.

Lemma 3.8.4. A subset $H \subseteq[n]$ is a generalized independent set of $\mathcal{H}$ if and only if for all nonempty subsets $K \subseteq H$, there exists some $i \in[m]$ such that $f(i) \in K$ and $\mathcal{X}_{i} \cap K=\varnothing$.

Proof. By definition, $H$ is a generalized independent set of $\mathcal{H}$ if and only if every nonempty subset $K$ of $H$ belongs to

$$
\mathcal{J}(\mathcal{H})=\bigcup_{i \in[m]}\left\{\{f(i)\} \cup Y_{i}: Y_{i} \subset \mathcal{Y}_{i}\right\}
$$

Equivalently, for every nonempty subset $K$ of $H$, there exists $i \in[m]$ such that $f(i) \in K$ and $K \backslash\{i\} \subseteq \mathcal{Y}_{i}$. In other words, for such a $K$, there exists $i \in[m]$ such that $f(i) \in K$ and $\mathcal{X}_{i} \cap K=\varnothing$.

The following lemma generalizes the second statement in [3, Lemma 19].

Lemma 3.8.5. If $H$ is a generalized independent set of $\mathcal{H}$, then the set of vertices

$$
V(H)=\left\{\boldsymbol{x} \in \mathbb{F}_{q}^{n}: \boldsymbol{x} \triangleleft H\right\}
$$

forms a clique in the confusion graph $\mathfrak{C}(\mathcal{H})$.
Proof. It suffices to show that for any $\boldsymbol{x} \triangleleft H$ and $\boldsymbol{y} \triangleleft H$, where $\boldsymbol{x} \neq \boldsymbol{y}$, it holds that $\boldsymbol{x}$ and $\boldsymbol{y}$ are adjacent in $\mathfrak{C}(\mathcal{H})$.

Let $T=\left\{j: x_{j}=y_{j}\right\}$ and $S=\operatorname{supp}(\boldsymbol{x}) \cup \operatorname{supp}(\boldsymbol{y})$. Then $K=S \backslash T \neq \varnothing$ since $\boldsymbol{x} \neq \boldsymbol{y}$. As $K \subseteq H$, by Lemma 3.8.4, there exists some $i \in[m]$ such that $f(i) \in K$ and $\mathcal{X}_{i} \cap K=\varnothing$. Since $f(i) \in K$, we have $x_{f(i)} \neq y_{f(i)}$. On the other hand, since $\mathcal{X}_{i} \cap K=\varnothing$, by definition of $K$, we have $x_{j}=y_{j}$ for all $j \in \mathcal{X}_{i}$. Hence, $\boldsymbol{x}$ and $\boldsymbol{y}$ are adjacent in $\mathfrak{C}(\mathcal{H})$.

The $\alpha$-bound for nonlinear ECIC's is stated as follows.

Proposition 3.8.6. The length of an optimal $(\delta, \mathcal{H})$-ECIC over $\mathbb{F}_{q}$ satisfies

$$
\mathcal{N}_{q}(\delta, \mathcal{H}) \geq N_{q}\left(q^{\alpha(\mathcal{H})}, 2 \delta+1\right)
$$

where $N_{q}(M, d)$ denotes the length of a shortest code of size $M$ and distance $d$.
Proof. Let $\mathfrak{E}$ be an optimal $(\delta, \mathcal{H})$-ECIC over $\mathbb{F}_{q}$ of length $N=\mathcal{N}_{q}(\delta, \mathcal{H})$. Let $H$ be a maximum generalized independent set of $\mathcal{H}$. Then $|H|=\alpha(\mathcal{H})$. By Lemma 3.8.5, the set $V(H)=\left\{\boldsymbol{x} \in \mathbb{F}_{q}^{n}: \boldsymbol{x} \triangleleft H\right\}$ forms a clique in $\mathcal{H}$. Moreover $|V(H)|=q^{\alpha(\mathcal{H})}$. By Lemma 3.8.3, the set $\mathscr{C}(H) \triangleq\{\mathfrak{E}(\boldsymbol{x}): \boldsymbol{x} \in V(H)\}$ forms an $\left(N, q^{\alpha(\mathcal{H})}, 2 \delta+1\right)_{q}$ code. Therefore,

$$
N \geq N_{q}\left(q^{\alpha(\mathcal{H})}, 2 \delta+1\right)
$$

The proof follows.

Remark 3.8.7. This lower bound for general ECIC's is certainly less than or equal to the respective bound for linear ECIC's, which is $N_{q}[\alpha(\mathcal{H}), 2 \delta+1]$.

The Singleton bound for nonlinear ECIC's is stated as follows.

Proposition 3.8.8. The length of an optimal $(\delta, \mathcal{H})$-ECIC over $\mathbb{F}_{q}$ satisfies

$$
\mathcal{N}_{q}(\delta, \mathcal{H}) \geq \kappa_{q}^{*}+2 \delta
$$

where $\kappa_{q}^{*}$ is the length of an optimal $\mathcal{H}-I C$ over $\mathbb{F}_{q}$.
Proof. Let $\mathfrak{E}$ be an optimal $(\delta, \mathcal{H})$-ECIC over $\mathbb{F}_{q}$ of length $N=\mathcal{N}_{q}(\delta, \mathcal{H})$. By Lemma 3.8.3,

$$
\begin{equation*}
\mathrm{d}\left(\mathfrak{E}(\boldsymbol{x}), \mathfrak{E}\left(\boldsymbol{x}^{\prime}\right)\right) \geq 2 \delta+1, \tag{3.22}
\end{equation*}
$$

for every pair of confusable vectors $\boldsymbol{x}$ and $\boldsymbol{x}^{\prime}$ in $\mathbb{F}_{q}^{n}$. Let $\mathfrak{E}^{d}: \mathbb{F}_{q}^{n} \rightarrow \mathbb{F}_{q}^{N-2 \delta}$ be obtained from $\mathfrak{E}$ as follows

$$
\left(\mathfrak{E}^{d}(\boldsymbol{x})\right)_{i}=(\mathfrak{E}(\boldsymbol{x}))_{i}, i \in[N-2 \delta] .
$$

In other words, the values of $\mathfrak{E}^{d}$ is obtained from the corresponding values of $\mathfrak{E}$ by discarding the last $2 \delta$ coordinates. Then from (3.22) we have

$$
\mathrm{d}\left(\mathfrak{E}^{d}(\boldsymbol{x}), \mathfrak{E}^{d}\left(\boldsymbol{x}^{\prime}\right)\right) \geq 1,
$$

for every pair of confusable vectors $\boldsymbol{x}$ and $\boldsymbol{x}^{\prime}$ in $\mathbb{F}_{q}^{n}$. Again by Lemma 3.8.3, $\mathfrak{E}^{d}$ is an $\mathcal{H}$-IC over $\mathbb{F}_{q}$. Therefore, its length is bounded from below by $\kappa_{q}^{*}$. Thus, $N-2 \delta \geq \kappa_{q}^{*}$,
which concludes the proof.

An immediate corollary of this bound is that for sufficiently large $q$, an optimal ECIC can be obtained by applying an MDS code on top of an optimal nonlinear IC.

Corollary 3.8.9. For $q \geq \kappa_{q}^{*}+2 \delta-1$,

$$
\mathcal{N}_{q}(\delta, \mathcal{H})=\kappa_{q}^{*}+2 \delta
$$

where $\kappa_{q}^{*}$ is the length of an optimal $\mathcal{H}-I C$ over $\mathbb{F}_{q}$.

There has been very limited knowledge on the length $\kappa_{q}^{*}$ of an optimal $\mathcal{H}$-IC over $\mathbb{F}_{q}$. It was shown by Bar-Yossef et al. $[3,4]$ (in the original setting of the ICSI problem) that $\kappa_{q}^{*}$ is bounded from below by $\alpha(\mathcal{D})$ where $D$ is the corresponding side information (di)graph. Therefore, for ICSI instances whose side information (di)graphs $\mathcal{D}$ satisfy $\alpha(\mathcal{D})=\operatorname{minrk}_{q}(\mathcal{D})$, we have $\kappa_{q}^{*}=\alpha(\mathcal{D})=\operatorname{minrk}_{q}(\mathcal{D})$. For such instances, there exist optimal ECIC's which are linear. The instances described by perfect graphs and acyclic digraphs are the only known families of ICSI instances which possess this special feature [3]. In Chapter 5, several new families of such instances are discovered.

## 4. (DI)GRAPHS OF EXTREME MINRANKS

The minrank of a graph was introduced by Haemers [34] to bound the Shannon capacity [57] of a graph. This parameter of a digraph has recently drawn much more attention from the research community after the work of Bar-Yossef et al. [3]. In their paper, it was shown that the binary minrank of a digraph $\mathcal{D}$ characterizes the length of an optimal binary scalar linear index code for an ICSI instance described by the digraph $\mathcal{D}$. A generalization of this result for an arbitrary finite field $\mathbb{F}_{q}$ is straightforward. Since a graph can be regarded as a symmetric digraph, the same result also holds for graphs. In this chapter, we characterize the (di)graphs that have extreme minranks. Based on these characterizations, it is shown in the next chapter that the problem of deciding whether the minrank of a digraph is equal to two is NP-complete. In contrast, the same question for graphs can be answered in polynomial time. In the context of index coding, we only study minranks of (di)graphs over a finite field $\mathbb{F}_{q}$. However, all of our results presented in Chapter 4 and Chapter 5 (except Theorem 4.2.4, Corollary 4.2.5, and Corollary 5.1.2) still hold for an arbitrary field $\mathbb{F}$. This is because the characteristic of the field does not play any role in their proofs.

## 4.1 (Strongly) Connected Components and MinRanks

Lemma 4.1.1 (Folklore). Let $\mathcal{G}=(\mathcal{V}(\mathcal{G}), \mathcal{E}(\mathcal{G}))$ be a graph. Suppose that $\mathcal{G}_{1}, \mathcal{G}_{2}, \ldots, \mathcal{G}_{k}$ are subgraphs of $\mathcal{G}$ that satisfy the following conditions

1. The sets $\mathcal{V}\left(\mathcal{G}_{i}\right), i \in[k]$, partition $\mathcal{V}(\mathcal{G})$;
2. There is no edge of the form $\{u, v\}$ where $u \in \mathcal{V}\left(\mathcal{G}_{i}\right)$ and $v \in \mathcal{V}\left(\mathcal{G}_{j}\right)$ for $i \neq j$.

Then

$$
\operatorname{minrk}_{q}(\mathcal{G})=\sum_{i=1}^{k} \operatorname{minrk}_{q}\left(\mathcal{G}_{i}\right)
$$

In particular, the above equality holds if $\mathcal{G}_{1}, \mathcal{G}_{2}, \ldots, \mathcal{G}_{k}$ are all connected components of $\mathcal{G}$.

Proof. The proof follows directly from the fact that a matrix fits $\mathcal{G}$ if and only if it is a block diagonal matrix (relabeling the vertices if necessary), the blocks of which fit the corresponding subgraphs $\mathcal{G}_{i}$ 's, $i \in[k]$.

Lemma 4.1.2 (Folklore). Let $\mathcal{D}=(\mathcal{V}(\mathcal{D}), \mathcal{E}(\mathcal{D}))$ be a digraph. If $\mathcal{D}_{1}, \mathcal{D}_{2}, \ldots, \mathcal{D}_{k}$ are all strongly connected components of $\mathcal{D}$, then

$$
\operatorname{minrk}_{q}(\mathcal{D})=\sum_{i=1}^{k} \operatorname{minrk}_{q}\left(\mathcal{D}_{i}\right)
$$

Proof. Suppose that $\mathcal{V}_{i}$ is the set of vertices that induces $\mathcal{D}_{i}, i \in[k]$. Then $\left\{\mathcal{V}_{i}\right\}_{i \in[k]}$ forms a partition of $\mathcal{V}(\mathcal{D})$. Relabeling the vertices of $\mathcal{D}$ if necessary, we may assume without loss of generality that for every $i<j$

1. $v_{i}<v_{j}$ whenever $v_{i} \in \mathcal{V}_{i}$ and $v_{j} \in \mathcal{V}_{j}$;
2. There are no $\operatorname{arcs}$ of the form $\left(v_{j}, v_{i}\right)$ where $v_{i} \in \mathcal{D}_{i}$ and $v_{j} \in \mathcal{D}_{j}$.

If $\boldsymbol{M}_{i}$ is a matrix that fits $\mathcal{D}_{i}, i \in[k]$, then the diagonal block matrix $\boldsymbol{M}$ whose diagonal blocks are $\boldsymbol{M}_{i}$ clearly fits $\mathcal{D}$. Moreover, $\operatorname{rank}_{q}(\boldsymbol{M})=\sum_{i=1}^{k} \operatorname{rank}_{q}\left(\boldsymbol{M}_{i}\right)$. Hence $\operatorname{minrk}_{q}(\mathcal{D}) \leq \sum_{i=1}^{k} \operatorname{minrk}_{q}\left(\mathcal{D}_{i}\right)$.

It remains to show that $\operatorname{minrk}_{q}(\mathcal{D}) \geq \sum_{i=1}^{k} \operatorname{minrk}_{q}\left(\mathcal{D}_{i}\right)$. Suppose that the matrix $\boldsymbol{M}$ fits $\mathcal{D}$. Then $\boldsymbol{M}$ is an upper-triangular block matrix, as shown in Fig. 4.1. If we let $\boldsymbol{M}_{i, i}$ be the sub-matrix of $\boldsymbol{M}$ formed by the rows and columns indexed by the elements of $\mathcal{V}_{i}$, then $\boldsymbol{M}_{i, i}$ fits $\mathcal{D}_{i}$ and hence,

$$
\operatorname{rank}_{q}(\boldsymbol{M}) \geq \sum_{i=1}^{k} \operatorname{rank}_{q}\left(\boldsymbol{M}_{i, i}\right) \geq \sum_{i=1}^{k} \operatorname{minrk}_{q}\left(\mathcal{D}_{i}\right)
$$

Thus, $\operatorname{minrk}_{q}(\mathcal{D}) \geq \sum_{i=1}^{k} \operatorname{minrk}_{q}\left(\mathcal{D}_{i}\right)$.


Fig. 4.1: Matrix $M$ that fits $\mathcal{D}$

These two lemmas suggest that it is sufficient to study the minranks of connected graphs and strongly connected digraphs.

## 4.2 (Di)Graphs of Extreme MinRanks

In this section, we investigate the structural properties of (di)graphs that have extreme minranks.

### 4.2.1 (Di)Graphs of MinRank One

Lemma 4.2.1 (Folklore). Let $\mathcal{D}=(\mathcal{V}(\mathcal{D}), \mathcal{E}(\mathcal{D}))$ be a digraph. Then $\operatorname{minrk}_{q}(\mathcal{D})=1$ if and only if $\mathcal{D}$ is a complete digraph. The same statement holds for a graph.

Proof. Suppose $\mathcal{D}$ is a digraph. If $\operatorname{minrk}_{q}(\mathcal{D})=1$, by definition there exists a matrix $\boldsymbol{M}=\left(m_{i, j}\right)$ of rank one over $\mathbb{F}_{q}$ that fits $\mathcal{D}$. Then the rows of $\boldsymbol{M}$ must be scalar multiples of each other. Moreover, by definition, $a_{i, i} \neq 0$ for all $i \in \mathcal{V}(\mathcal{D})$. Hence $a_{i, j} \neq 0$ for all $i \in \mathcal{V}(\mathcal{D})$ and all $j \in \mathcal{V}(\mathcal{D})$. Therefore, $(i, j) \in \mathcal{E}(\mathcal{D})$ for all $i \neq j$, $i \in \mathcal{V}(\mathcal{D})$ and $j \in \mathcal{V}(\mathcal{D})$. In other words, $\mathcal{D}$ is a complete digraph.

Conversely, suppose that $\mathcal{D}$ is a complete digraph. Then $\boldsymbol{J}$, the all-one matrix, fits $\mathcal{D}$ and $\operatorname{minrk}_{q}(\boldsymbol{J})=1$, which implies that $\operatorname{minrk}_{q}(\mathcal{D})=1$. The same arguments hold for graphs.

### 4.2.2 (Di)Graphs of MinRank Two

We first study (di)graphs of minranks at most two. In this section, only binary alphabet is considered. We first introduce the following concept of a fair coloring of a digraph. Recall that a $k$-coloring of a graph $\mathcal{G}=(\mathcal{V}(\mathcal{G}), \mathcal{E}(\mathcal{G}))$ is a mapping $\phi$ : $\mathcal{V}(\mathcal{G}) \rightarrow[k]$ which satisfies the condition that $\phi(u) \neq \phi(v)$ whenever $\{u, v\} \in \mathcal{E}(\mathcal{G})$. We often refer to $\phi(u)$ as the color of $u$. If there exists a $k$-coloring of $\mathcal{G}$, then we say that $\mathcal{G}$ is $k$-colorable.

Definition 4.2.2. Let $\mathcal{D}=(\mathcal{V}(\mathcal{D}), \mathcal{E}(\mathcal{D}))$ be a digraph. A fair $k$-coloring of $\mathcal{D}$ is a mapping $\phi: \mathcal{V}(\mathcal{D}) \rightarrow[k]$ that satisfies the following conditions:
(C1) If $(u, v) \in \mathcal{E}(\mathcal{D})$ then $\phi(u) \neq \phi(v)$;
(C2) For each vertex $u$ of $\mathcal{D}$, it holds that $\phi(v)=\phi(\omega)$ for all out-neighbors $v$ and $\omega$ of $u$.

If there exists a fair $k$-coloring of $\mathcal{D}$, we say that we can color $\mathcal{D}$ fairly by $k$ colors, or, $\mathcal{D}$ is fairly $k$-colorable.

We refer to the condition (C2) as the fairness of the coloring, since this condition guarantees that all out-neighbors of each vertex share the same color.

Lemma 4.2.3. A digraph $\mathcal{D}=(\mathcal{V}(\mathcal{D}), \mathcal{E}(\mathcal{D}))$ is fairly 3-colorable if and only if there exists a partition of $\mathcal{V}(\mathcal{D})$ into three subsets $A, B$, and $C$ that satisfy the following conditions:

1. For every $i \in A$ : either $N_{O}(i) \subseteq B$ or $N_{O}(i) \subseteq C$;
2. For every $i \in B$ : either $N_{O}(i) \subseteq A$ or $N_{O}(i) \subseteq C$;
3. For every $i \in C$ : either $N_{O}(i) \subseteq A$ or $N_{O}(i) \subseteq B$.

Proof. If $\mathcal{D}$ is fairly 3 -colorable, let $A, B$, and $C$ respectively be the sets of vertices of $\mathcal{D}$ that share the same color. Then clearly $A, B$, and $C$ partition $\mathcal{V}(\mathcal{D})$. Moreover, since all out-neighbors of each vertex must have the same color, the three conditions above are obviously satisfied. Conversely, if those conditions are satisfied, then $\phi: \mathcal{V}(\mathcal{D}) \rightarrow[3]$ defined by

$$
\phi(u)= \begin{cases}1, & u \in A \\ 2, & u \in B \\ 3, & u \in C\end{cases}
$$

is a fair 3 -coloring of $\mathcal{D}$.

Theorem 4.2.4. Let $\mathcal{D}=(\mathcal{V}(\mathcal{D}), \mathcal{E}(\mathcal{D}))$ be a digraph. Then $\operatorname{minrk}_{2}(\mathcal{D}) \leq 2$ if and only if $\overline{\mathcal{D}}$, the complement of $\mathcal{D}$, is fairly 3 -colorable.

Proof. Suppose $\mathcal{V}(\mathcal{D})=[n]$. By definition, $\operatorname{minrk}_{2}(\mathcal{D}) \leq 2$ if and only if there exists an $n \times n$ binary matrix $\boldsymbol{M}$ of rank at most two which fits $\mathcal{D}$. The matrix $\boldsymbol{M}$ has rank less than or equal to two if and only if there are two rows of $\boldsymbol{M}$ that span its row space. Without loss of generality, suppose that they are the first two rows of $\boldsymbol{M}$, namely, $\boldsymbol{M}_{1}$ and $\boldsymbol{M}_{2}$. Let

$$
B=\operatorname{supp}\left(\boldsymbol{M}_{1}\right) \cap \operatorname{supp}\left(\boldsymbol{M}_{2}\right)
$$

and

$$
A=\operatorname{supp}\left(\boldsymbol{M}_{1}\right) \backslash B, \quad C=\operatorname{supp}\left(\boldsymbol{M}_{2}\right) \backslash B
$$

Since the binary alphabet is considered, for every $i \in[n]$, one of the following must hold: (1) $\boldsymbol{M}_{i}=\boldsymbol{M}_{1} ;(2) \boldsymbol{M}_{i}=\boldsymbol{M}_{2} ;(3) \boldsymbol{M}_{i}=\boldsymbol{M}_{1}+\boldsymbol{M}_{2}$. Hence for every $i \in[n]$

$$
i \in \operatorname{supp}\left(\boldsymbol{M}_{i}\right) \subseteq A \cup B \cup C
$$

This implies that $A \cup B \cup C=[n]$.
Suppose that $i \in A$. Then either $\boldsymbol{M}_{i}=\boldsymbol{M}_{1}$ or $\boldsymbol{M}_{i}=\boldsymbol{M}_{1}+\boldsymbol{M}_{2}$. The former holds if and only if $\operatorname{supp}\left(\boldsymbol{M}_{i}\right)=A \cup B$. Equivalently, we obtain that $(i, j) \in \mathcal{E}(\mathcal{D})$ for all $j \in A \cup B \backslash\{i\}$. In other words, $(i, j) \notin \mathcal{E}(\overline{\mathcal{D}})$ for all $j \in A \cup B$. The latter holds if and only if $\operatorname{supp}\left(\boldsymbol{M}_{i}\right)=A \cup C$. Similarly, this equality in turn is equivalent to the property that $(i, j) \notin \mathcal{E}(\overline{\mathcal{D}})$ for all $j \in A \cup C$. In summary, for every $i \in A$

1. $(i, j) \notin \mathcal{E}(\overline{\mathcal{D}})$, for all $j \in A$;
2. Either $(i, j) \notin \mathcal{E}(\overline{\mathcal{D}})$, for all $j \in B$, or $(i, j) \notin \mathcal{E}(\overline{\mathcal{D}})$, for all $j \in C$; In other words, either $\mathcal{N}_{O}^{\bar{D}}(i) \subseteq B$ or $\mathcal{N}_{O}^{\bar{D}}(i) \subseteq C$.

Analogous conditions hold for every $i \in B$ and for every $i \in C$ as well. Therefore, by Lemma $4.2 .3, \operatorname{minrk}_{2}(\mathcal{D}) \leq 2$ if and only if $\overline{\mathcal{D}}$ is fairly 3 -colorable.

The following corollary characterizes the digraphs of minrank two.

Corollary 4.2.5. A digraph $\mathcal{D}$ has minrank two over $\mathbb{F}_{2}$ if and only if $\overline{\mathcal{D}}$ is fairly 3-colorable and $\mathcal{D}$ is not a complete digraph.

### 4.2.3 (Di)Graphs of MinRanks Equal to Their Orders

For graphs, it is simple to characterize graphs of minranks equal to their orders.

Proposition 4.2.6. Let $\mathcal{G}$ be a graph of order $n$. Then $\operatorname{minrk}_{q}(\mathcal{G})=n$ if and only if $\mathcal{G}$ has no edges.

Proof. If $\mathcal{G}$ has no edges, a matrix fits $\mathcal{G}$ if and only if it is a diagonal matrix whose entries on the main diagonal are all nonzero. The rank of such a matrix is $n$. Therefore, $\operatorname{minrk}_{q}(\mathcal{G})=n$. Suppose that $\operatorname{minrk}_{q}(\mathcal{G})=n$ and $(i, j)$ is an edge of $\mathcal{G}$. Then the matrix $\boldsymbol{M}$, where $\boldsymbol{M}_{k}=\boldsymbol{e}_{k}$ for all $k \notin\{i, j\}$ and $\boldsymbol{M}_{i}=\boldsymbol{M}_{j}=\boldsymbol{e}_{i}+\boldsymbol{e}_{j}$, fits $\mathcal{G}$. Moreover, $\operatorname{rank}_{q}(\boldsymbol{M})=n-1$. Hence, $\operatorname{minrk}_{q}(\mathcal{G}) \leq n-1$. We obtain a contradiction.

For digraphs, the characterization is not that obvious.

Theorem 4.2.7. Let $\mathcal{D}$ be a digraph of order $n$. Then $\operatorname{minrk}_{q}(\mathcal{D})=n$ if and only if $\mathcal{D}$ is acyclic.

Proof. Equivalently, we show that $\operatorname{minrk}_{q}(\mathcal{D}) \leq n-1$ if and only if $\mathcal{D}$ has a circuit.
Suppose without loss of generality that $\mathcal{C}=(1,2, \ldots, \ell)$ is a circuit in $\mathcal{D}$. We construct a matrix $\boldsymbol{M}$ fitting $\mathcal{D}$ as follows. For $j \notin \mathcal{V}(\mathcal{C})$, let $\boldsymbol{M}_{j}=\boldsymbol{e}_{j}$. For $i \in[\ell-1]$, let $\boldsymbol{M}_{i}=\boldsymbol{e}_{i}-\boldsymbol{e}_{i+1}$. Finally, let $\boldsymbol{M}_{\ell}=\boldsymbol{e}_{1}-\boldsymbol{e}_{\ell}$. Clearly, $\boldsymbol{M}$ fits $\mathcal{D}$. Moreover, as $\boldsymbol{M}_{\ell}=\sum_{i=1}^{\ell-1} \boldsymbol{M}_{i}$, we have $\operatorname{rank}_{q}\left(\boldsymbol{M}_{\mathcal{V}(\mathcal{C})}\right)=\ell-1$. Hence

$$
\begin{aligned}
\operatorname{rank}_{q}(\boldsymbol{M}) & =\operatorname{rank}_{q}\left(\boldsymbol{M}_{\mathcal{V}(\mathcal{C})}\right)+\operatorname{rank}_{q}\left(\boldsymbol{M}_{[n] \backslash \mathcal{V}(\mathcal{C})}\right) \\
& =(\ell-1)+(n-\ell) \\
& =n-1 .
\end{aligned}
$$

Therefore, $\operatorname{minrk}_{q}(\mathcal{D}) \leq n-1$.
Conversely, suppose that $\operatorname{minrk}_{q}(\mathcal{D}) \leq n-1$. Then there exists a matrix $\boldsymbol{M}$ fitting $\mathcal{D}$ whose rows are linearly dependent. In other words, $\sum_{i \in I} \alpha_{i} \boldsymbol{M}_{i}=\mathbf{0}$ for some nonempty subset $I \subseteq[n]$ and for some $\alpha_{i} \in \mathbb{F}_{q}^{*}, i \in I$. Let $\mathcal{D}^{\prime}$ be the subgraph of $\mathcal{D}$ induced by the vertices in $I$ and $\boldsymbol{M}^{\prime}$ the sub-matrix of $\boldsymbol{M}$ restricted to the rows and columns indexed by the elements of $I$. Obviously $\boldsymbol{M}^{\prime}$ fits $\mathcal{D}^{\prime}$. We show that there exists a circuit in $\mathcal{D}^{\prime}$. Since $\sum_{i \in I} \alpha_{i} \boldsymbol{M}_{i}^{\prime}=\mathbf{0}$, each column of $\boldsymbol{M}^{\prime}$ has at least two nonzero entries. Therefore, for each vertex $v$ of $\mathcal{D}^{\prime}$, there exists another vertex $u$ of $\mathcal{D}^{\prime}$ such that $(u, v)$ is an arc in $\mathcal{D}^{\prime}$. Starting from an arbitrary vertex $v_{1}$ of $\mathcal{D}^{\prime}$ and applying this property recursively, we obtain a sequence of vertices in $\mathcal{D}^{\prime}$

$$
v_{1}, v_{2}, \ldots, v_{s}, v_{s+1}, \ldots
$$

where $\left(v_{s+1}, v_{s}\right)$ is an arc in $\mathcal{D}^{\prime}$ for every $s \geq 1$. Since $\mathcal{D}^{\prime}$ has a finite number of vertices, there must be a time (we care about the first time only) when a vertex appears twice in the above sequence. This vertex, together with the other vertices lying between its two occurrences, form a circuit inside $\mathcal{D}^{\prime}$, which is also a circuit
inside $\mathcal{D}$.

The existence of a circuit in a digraph can be detected by using a depth-first search, the time complexity of which is polynomial with respect to the order of the digraph. Hence, as a consequence of Theorem 4.2.7, the problem of deciding whether a digraph has minrank equal to its order can be solved in polynomial time.

### 4.2.4 Graphs of MinRanks One Less Than Their Orders

We only consider graphs.

Definition 4.2.8. A graph $\mathcal{G}=(\mathcal{V}(\mathcal{G}), \mathcal{E}(\mathcal{G}))$ is called a star graph if $|\mathcal{V}(\mathcal{G})| \geq 2$ and there exists a vertex $v \in \mathcal{V}(\mathcal{G})$ such that $\mathcal{E}(\mathcal{G})=\{\{u, v\}: u \in \mathcal{V}(\mathcal{G}) \backslash\{v\}\}$.


Fig. 4.2: A star graph

Theorem 4.2.9. Let $\mathcal{G}$ be a connected graph of order $n \geq 2$. Then $\operatorname{minrk}_{q}(\mathcal{G})=n-1$ if and only if $\mathcal{G}$ is a star graph.

Proof. We first suppose that $\operatorname{minrk}_{q}(\mathcal{G})=n-1$. If $n=2$ then $\mathcal{G}$ must be a complete graph, which is also a star graph. We assume that $n \geq 3$. As $\mathcal{G}$ is connected, there exists a vertex $v$ of degree at least two. Let $v_{1}$ and $v_{2}$ be any two distinct vertices adjacent to $v$. Our goal is to show that for every vertex $u \neq v$, we have $\{u, v\} \in \mathcal{E}(\mathcal{G})$, and those are all possible edges.

1. Suppose for a contradiction that $\{u, v\} \notin \mathcal{E}(\mathcal{G})$ for some $u \neq v$. Since $\mathcal{G}$ is connected, there exists $\omega$ such that $\{u, \omega\} \in \mathcal{E}(\mathcal{G})$. Then either $\omega \neq v_{1}$ or $\omega \neq v_{2}$. Suppose that $\omega \neq v_{2}$. We create a matrix $\boldsymbol{M}$ as follows. Let $\boldsymbol{M}_{u}=\boldsymbol{M}_{\omega}=\boldsymbol{e}_{u}+\boldsymbol{e}_{\omega}, \boldsymbol{M}_{v}=\boldsymbol{M}_{v_{2}}=\boldsymbol{e}_{v}+\boldsymbol{e}_{v_{2}}$, and $\boldsymbol{M}_{i}=\boldsymbol{e}_{i}$ for $i \notin\left\{u, v, v_{2}, \omega\right\}$. Then $\boldsymbol{M}$ fits $\mathcal{G}$ and $\operatorname{rank}_{q}(\boldsymbol{M})=n-2$. Therefore, $\operatorname{minrk}_{q}(\mathcal{G})<n-1$. We obtain a contradiction. Thus, all other vertices are adjacent to $v$.
2. Suppose for a contradiction that there exist two adjacent vertices, namely $u$ and $\omega$, both are different from $v$. As we just proved, both $u$ and $\omega$ must be adjacent to $v$. We create a matrix $\boldsymbol{M}$ as follows. We take $\boldsymbol{M}_{u}=\boldsymbol{M}_{v}=$ $\boldsymbol{M}_{\omega}=\boldsymbol{e}_{u}+\boldsymbol{e}_{v}+\boldsymbol{e}_{\omega}$, and $\boldsymbol{M}_{i}=\boldsymbol{e}_{i}$ for $i \notin\{u, v, \omega\}$. Clearly $\boldsymbol{M}$ fits $\mathcal{G}$ and moreover, $\operatorname{rank}_{q}(\boldsymbol{M})=n-2$, which implies that $\operatorname{minrk}_{q}(\mathcal{G})<n-1$. We obtain a contradiction.

Conversely, assume that $\mathcal{G}$ is a star graph, where $\mathcal{E}(\mathcal{G})=\left\{\left\{v, v_{i}\right\}: i \in[n-1]\right\}$, $v_{i} \in \mathcal{V}(\mathcal{G}) \backslash\{v\}$ for all $i \in[n-1]$. We create a matrix $\boldsymbol{M}$ fitting $\mathcal{G}$ by taking $\boldsymbol{M}_{v_{i}}=\boldsymbol{e}_{v}+\boldsymbol{e}_{v_{i}}$ for $i \in[n-1]$, and $\boldsymbol{M}_{v}=\boldsymbol{e}_{v}+\boldsymbol{e}_{v_{1}}$. Since $\boldsymbol{M}_{v} \equiv \boldsymbol{M}_{v_{1}}$, we deduce that $\operatorname{rank}_{q}(\boldsymbol{M})=n-1$. Hence $\operatorname{minrk}_{q}(\mathcal{G}) \leq n-1$. On the other hand, since $\left\{v_{i}: \quad i \in[n-1\}\right.$ is a maximum independent set in $\mathcal{G}$, we obtain that $\alpha(\mathcal{G})=n-1$. By Theorem 3.2.11, $\operatorname{minrk}_{q}(\mathcal{G}) \geq \alpha(\mathcal{G})=n-1$. Thus, $\operatorname{minrk}_{q}(\mathcal{G})=n-1$.

Corollary 4.2.10. Let $\mathcal{G}$ be a graph of order $n \geq 2$ and let $\mathcal{G}_{1}, \mathcal{G}_{2}, \ldots, \mathcal{G}_{k}$ be all of its connected components, with

$$
\left|\mathcal{V}\left(\mathcal{G}_{1}\right)\right| \geq\left|\mathcal{V}\left(\mathcal{G}_{2}\right)\right| \geq \cdots \geq\left|\mathcal{V}\left(\mathcal{G}_{k}\right)\right|
$$

Then $\operatorname{minrk}_{q}(\mathcal{G})=n-1$ if and only if the following conditions are satisfied

1. $\mathcal{G}_{1}$ is a star graph having at least two vertices,
2. $\mathcal{G}_{i}$ is a one-vertex graph for every $i \geq 2$.

Proof. Suppose that $\operatorname{minrk}_{q}(\mathcal{G})=n-1$. Then $\mathcal{G}$ must have some connected component that contains at least two vertices. Otherwise, by Proposition 4.2.6, $\operatorname{minrk}_{q}(\mathcal{G})$ would be $n$. Hence, as the largest component, $\mathcal{G}_{1}$ must contain at least two vertices. Suppose for a contradiction that there exists another connected component, say $\mathcal{G}_{2}$, that contains more than one vertex. As each of $\mathcal{G}_{1}$ and $\mathcal{G}_{2}$ contains at least one edge, by Lemma 4.1.1 and Proposition 4.2.6, we obtain

$$
\begin{aligned}
\operatorname{minrk}_{q}(\mathcal{G}) & =\operatorname{minrk}_{q}\left(\mathcal{G}_{1}\right)+\operatorname{minrk}_{q}\left(\mathcal{G}_{2}\right)+\sum_{i=3}^{k} \operatorname{minrk}_{q}\left(\mathcal{G}_{i}\right) \\
& \leq\left(\left|\mathcal{V}\left(\mathcal{G}_{1}\right)\right|-1\right)+\left(\left|\mathcal{V}\left(\mathcal{G}_{2}\right)\right|-1\right)+\sum_{i=3}^{k}\left|\mathcal{V}\left(\mathcal{G}_{i}\right)\right| \\
& <n-1
\end{aligned}
$$

We have a contradiction. Therefore, except from $\mathcal{G}_{1}$, all other connected components each contains precisely one vertex. Moreover, $\operatorname{minrk}_{q}\left(\mathcal{G}_{1}\right)=\left|\mathcal{V}\left(\mathcal{G}_{1}\right)\right|-1$, for otherwise,

$$
\begin{aligned}
\operatorname{minrk}_{q}(\mathcal{G}) & =\operatorname{minrk}_{q}\left(\mathcal{G}_{1}\right)+\sum_{i=2}^{k} \operatorname{minrk}_{q}\left(\mathcal{G}_{i}\right) \\
& \leq\left(\left|\mathcal{V}\left(\mathcal{G}_{1}\right)\right|-2\right)+\sum_{i=2}^{k}\left|\mathcal{V}\left(\mathcal{G}_{i}\right)\right| \\
& <n-1
\end{aligned}
$$

Therefore, by Theorem 4.2.9, $\mathcal{G}_{1}$ must be a star graph.
Conversely, suppose that $\mathcal{G}_{1}$ is a star graph with at least two vertices, and that
$\mathcal{G}_{i}$ for $i>1$ contains precisely one vertex. Then $\left|\mathcal{V}\left(\mathcal{G}_{1}\right)\right|=n-k+1$ and

$$
\begin{aligned}
\operatorname{minrk}_{q}(\mathcal{G}) & =\operatorname{minrk}_{q}\left(\mathcal{G}_{1}\right)+\sum_{i=2}^{k} \operatorname{minrk}_{q}\left(\mathcal{G}_{i}\right) \\
& =\left(\left|\mathcal{V}\left(\mathcal{G}_{1}\right)\right|-1\right)+(k-1) \\
& =n-1
\end{aligned}
$$

## 5. COMPUTATION OF MINRANKS OF SIDE INFORMATION (DI)GRAPHS

In this chapter, the computational aspects of the ICSI problem are investigated. Firstly, we show that deciding whether a digraph has minrank two is an NP-complete problem. For comparison, there is a polynomial time to decide if a graph has minrank two. Secondly, we establish a new upper bound for the minrank of a digraph, namely the circuit-packing bound, which is arguably more suitable for digraphs than the best currently known upper bound. Employing this new bound, we point out several families of digraphs whose minranks can be found in polynomial time. Moreover, for ICSI instances described by such digraphs, scalar linear index codes are shown to be optimal. Thirdly, a polynomial time dynamic programming algorithm is developed to compute the minranks of a family of graphs possessing a special tree structure. Intuitively, such graphs are obtained by gluing together, in a tree-like structure, several graphs for which the minranks can be determined in polynomial time. Finally, using a computer program, we compute the minranks of all nonisomorphic graphs of order up to 10 .

### 5.1 The Hardness of the MinRank Problem for Digraphs

In this section, we first prove that it is an NP-complete problem to decide whether a given digraph is fairly $k$-colorable (see Definition 4.2.2), for $k \geq 3$. The hardness
of this problem, by Lemma 4.2.1 and Corollary 4.2.5, leads to the hardness of the problem of deciding whether a given digraph has minrank two over $\mathbb{F}_{2}$. The fair $k$-coloring problem is defined formally as follows.

Problem: FAIR $\boldsymbol{k}$-COLORING
Instance: A digraph $\mathcal{D}$
Question: Is $\mathcal{D}$ fairly $k$-colorable?

Theorem 5.1.1. The fair $k$-coloring problem is $N P$-complete for $k \geq 3$.

Proof. This problem is obviously in NP. For NP-hardness, we reduce the $k$-coloring problem to the fair $k$-coloring problem. Suppose that $\mathcal{G}=(\mathcal{V}(\mathcal{G}), \mathcal{E}(\mathcal{G}))$ is an arbitrary graph. We aim to build a digraph $\mathcal{D}=(\mathcal{V}(\mathcal{D}), \mathcal{E}(\mathcal{D}))$ so that $\mathcal{G}$ is $k$-colorable if and only if $\mathcal{D}$ is fairly $k$-colorable. Suppose that $\mathcal{V}(\mathcal{G})=[n]$. For each $i \in[n]$, we build the following gadget, which is a digraph $\mathcal{D}_{i}=\left(\mathcal{V}_{i}, \mathcal{E}_{i}\right)$. The vertex set of $\mathcal{D}_{i}$ is

$$
\mathcal{V}_{i}=\{i\} \cup\left\{\omega_{i, j}: j \in N^{\mathcal{G}}(i)\right\},
$$

where $\omega_{i, j}$ are newly introduced vertices. We refer to $\omega_{i, j}$ as a clone (in $\mathcal{D}_{i}$ ) of the vertex $j \in[n]$. The arc set of $\mathcal{D}_{i}$ is

$$
\mathcal{E}_{i}=\left\{\left(\omega_{i, j}, i\right): j \in N^{\mathcal{G}}(i)\right\} .
$$

Let $N^{\mathcal{G}}(i)=\left\{i_{1}, i_{2}, \ldots, i_{n_{i}}\right\}$. Then $\mathcal{D}_{i}$ can be drawn as in Fig. 5.1.

Moreover, we also introduce $n$ new vertices, which are $p_{1}, p_{2}, \ldots, p_{n}$. The digraph $\mathcal{D}=(\mathcal{V}(\mathcal{D}), \mathcal{E}(\mathcal{D}))$ is built as follows. The vertex set of $\mathcal{D}$ is

$$
\mathcal{V}(\mathcal{D})=\left(\cup_{i=1}^{n} \mathcal{V}_{i}\right) \cup\left\{p_{1}, p_{2}, \ldots, p_{n}\right\}
$$



Fig. 5.1: Gadget $\mathcal{D}_{i}$ for each vertex $i$ of $\mathcal{G}$

The arc set of $\mathcal{D}$ is

$$
\mathcal{E}(\mathcal{D})=\left(\cup_{i=1}^{n} \mathcal{E}_{i}\right) \cup\left(\cup_{i=1}^{n} \mathcal{Q}_{i}\right)
$$

where $\mathcal{Q}_{i}$ consists of $\left(p_{i}, i\right)$ and the arcs that connect $p_{i}$ and all the clones of $i$. More formally,

$$
\mathcal{Q}_{i}=\left\{\left(p_{i}, i\right)\right\} \cup\left\{\left(p_{i}, \omega_{i^{\prime}, i}\right): i^{\prime} \in[n], i \in N^{\mathcal{G}}\left(i^{\prime}\right)\right\} .
$$

For example, if $\mathcal{G}$ is the graph in Fig. 5.2, then $\mathcal{D}$ is the digraph in Fig. 5.3.


Fig. 5.2: The graph $\mathcal{G}$

Our goal now is to show that $\mathcal{G}$ is $k$-colorable if and only if $\mathcal{D}$ is fairly $k$-colorable.
Suppose that $\mathcal{G}$ is $k$-colorable and $\phi_{\mathcal{G}}:[n] \rightarrow[k]$ is a $k$-coloring of $\mathcal{G}$. We consider the mapping $\phi_{\mathcal{D}}: \mathcal{V}(\mathcal{D}) \rightarrow[k]$ defined as follows

1. For every $i \in[n], \phi_{\mathcal{D}}(i) \triangleq \phi_{\mathcal{G}}(i)$;
2. If $i \in N^{\mathcal{G}}\left(i^{\prime}\right)$ then $\phi_{\mathcal{D}}\left(\omega_{i^{\prime}, i}\right) \triangleq \phi_{\mathcal{D}}(i)=\phi_{\mathcal{G}}(i)$, in other words, clones of $i$ have


Fig. 5.3: The digraph $\mathcal{D}$ built from the graph $\mathcal{G}$ in Fig. 5.2
the same color as $i$;
3. For every $i \in[n], \phi_{\mathcal{D}}\left(p_{i}\right)$ can be chosen arbitrarily, as long as it is different from $\phi_{\mathcal{D}}(i)$.

We claim that $\phi_{\mathcal{D}}$ is a fair $k$-coloring for $\mathcal{D}$. We first verify the condition (C1) (see Definition 4.2.2). It is straightforward from the definition of $\phi_{\mathcal{D}}$ that the endpoints of each of the arcs of the forms $\left(p_{i}, i\right)$ for $i \in[n]$, and $\left(p_{i}, \omega_{i^{\prime}, i}\right)$ for $i \in N^{\mathcal{G}}\left(i^{\prime}\right)$, have different colors. It remains to check if $i$ and $\omega_{i, j}$ for $j \in N^{\mathcal{G}}(i)$ have different colors. On the one hand, $\omega_{i, j}$ is a clone of $j$, and hence has the same color as $j$. In other words,

$$
\phi_{\mathcal{D}}\left(\omega_{i, j}\right)=\phi_{\mathcal{D}}(j)=\phi_{\mathcal{G}}(j) .
$$

On the other hand, since $j \in N^{\mathcal{G}}(i)$, we obtain that

$$
\phi_{\mathcal{G}}(j) \neq \phi_{\mathcal{G}}(i)=\phi_{D}(i) .
$$

Therefore, $\phi_{\mathcal{D}}\left(\omega_{i, j}\right) \neq \phi_{\mathcal{D}}(i)$ for all $i \in[n]$ and $j \in N^{\mathcal{G}}(i)$. Thus, (C1) is satisfied.
We now check if (C2) (see Definition 4.2.2) is also satisfied. The out-neighbors
of $p_{i}$ are $i$ and its clones $\omega_{i^{\prime}, i}\left(i \in N^{\mathcal{G}}\left(i^{\prime}\right)\right)$. These vertices have the same color in $\mathcal{D}$, namely $\phi_{\mathcal{G}}(i)$, by the definition of $\phi_{\mathcal{D}}$. Thus (C2) is also satisfied. Therefore $\phi_{\mathcal{D}}$ is a fair $k$-coloring of $\mathcal{D}$.

Conversely, suppose that $\phi_{\mathcal{D}}: \mathcal{V}(\mathcal{D}) \rightarrow[k]$ is a fair $k$-coloring of $\mathcal{D}$. Condition $(\mathrm{C} 2)$ guarantees that all clones of $i$ have the same color as $i$, namely, $\phi_{\mathcal{D}}\left(\omega_{i^{\prime}, i}\right)=\phi_{\mathcal{D}}(i)$ if $i \in N^{\mathcal{G}}\left(i^{\prime}\right)$. Therefore, by $(\mathrm{C} 1)$, if $\{i, j\} \in \mathcal{E}(\mathcal{G})$, that is, $j \in N^{\mathcal{G}}(i)$, then

$$
\phi_{\mathcal{D}}(i) \neq \phi_{\mathcal{D}}\left(\omega_{i, j}\right)=\phi_{\mathcal{D}}(j) .
$$

Hence, if we define $\phi_{\mathcal{G}}:[n] \rightarrow[k]$ by $\phi_{\mathcal{G}}(i)=\phi_{\mathcal{D}}(i)$ for all $i \in[n]$, then it is a $k$-coloring of $\mathcal{G}$. Thus $\mathcal{G}$ is $k$-colorable.

Finally, notice that the order of $\mathcal{D}$ is a polynomial with respect to the order of $\mathcal{G}$. More specifically, $\mathcal{V}(\mathcal{D})=2|\mathcal{V}(\mathcal{G})|+2|\mathcal{E}(\mathcal{G})|$ and $\mathcal{E}(\mathcal{D})=|\mathcal{V}(\mathcal{G})|+4|\mathcal{E}(\mathcal{G})|$. Moreover, building $\mathcal{D}$ from $\mathcal{G}$, and also obtaining a coloring of $\mathcal{G}$ from a coloring of $\mathcal{D}$, can be done in polynomial time with respect to the order of $\mathcal{G}$. Since the $k$-coloring problem ( $k \geq 3$ ) is NP-hard [40], we conclude that the fair $k$-coloring problem is also NP-hard.

Corollary 5.1.2. Given an arbitrary digraph $\mathcal{D}$, deciding whether $\operatorname{minrk}_{2}(\mathcal{D})=2$ is $N P$-complete.

In contrast, for a graph $\mathcal{G}$, it was observed by Peeters [53] that $\mathcal{G}$ has minrank two if and only if $\overline{\mathcal{G}}$ is a bipartite graph and $\mathcal{G}$ is not a complete graph, which can be verified in polynomial time. This fact can also be derived by applying Theorem 4.2.4 to $\mathcal{D}^{\mathcal{G}}$, the digraph corresponding to $\mathcal{G}\left(\mathcal{D}^{\mathcal{G}}\right.$ is obtained from $\mathcal{G}$ by replacing each edge of $\mathcal{G}$ by two opposite arcs). Indeed, we first observe that a symmetric digraph is fairly 3 -colorable if and only if it is fairly 2 -colorable. Therefore, a symmetric
digraph is fairly 3 -colorable if and only if its underlying graph is 2 -colorable, i.e., is a bipartite graph. Hence, by Theorem 4.2.4, $\operatorname{minrk}_{2}\left(\mathcal{D}^{\mathcal{G}}\right) \leq 2$ if and only if $\overline{\mathcal{G}}$ is bipartite. As $\mathcal{G}$ and $\mathcal{D}^{\mathcal{G}}$ have the same minrank, we conclude that $\operatorname{minrk}_{2}(\mathcal{G}) \leq 2$ if and only if $\overline{\mathcal{G}}$ is bipartite.

### 5.2 Circuit-Packing Bound

In this section we introduce a new upper bound for the minrank of a digraph. This bound reveals some new families of digraphs whose minranks are computable in polynomial time.

### 5.2.1 The Bound

We first introduce an easy lemma.

Lemma 5.2.1. If $\mathcal{D}^{\prime}$ is a subgraph of a digraph $\mathcal{D}$ then

$$
\operatorname{minrk}_{q}(\mathcal{D}) \geq \operatorname{minrk}_{q}\left(\mathcal{D}^{\prime}\right)
$$

The same conclusion also holds for graphs.

Proof. Let $\boldsymbol{M}$ be a matrix that fits $\mathcal{D}$ and has rank equal to the minrank of $\mathcal{D}$. Then the sub-matrix $\boldsymbol{M}^{\prime}$ of $\boldsymbol{M}$ restricted on the rows and columns indexed by the vertices in $\mathcal{V}\left(\mathcal{D}^{\prime}\right)$ is a matrix that fits $\mathcal{D}^{\prime}$. Then

$$
\operatorname{minrk}_{q}\left(\mathcal{D}^{\prime}\right) \leq \operatorname{rank}_{q}\left(\boldsymbol{M}^{\prime}\right) \leq \operatorname{rank}_{q}(\boldsymbol{M})=\operatorname{minr}_{q}(\mathcal{D})
$$

The following lower bound on the minrank of a digraph is an immediate corollary of Theorem 4.2.7, which was also established by Bar-Yossef et al. [3, 4].

Corollary 5.2.2. For a digraph $\mathcal{D}$ we have

$$
\operatorname{minrk}_{q}(\mathcal{D}) \geq \alpha(\mathcal{D})
$$

where $\alpha(\mathcal{D})$ denotes the order of a maximum acyclic induced subgraph of $\mathcal{D}$.

Proof. Let $\mathcal{D}^{\prime}$ be a maximum acyclic induced subgraph of $\mathcal{D}$ of order $\alpha(\mathcal{D})$. Since $\mathcal{D}^{\prime}$ is acyclic, by Lemma 5.2.1 and Theorem 4.2.7 we have

$$
\operatorname{minrk}_{q}(\mathcal{D}) \geq \operatorname{minrk}_{q}\left(\mathcal{D}^{\prime}\right)=\alpha(\mathcal{D})
$$

Let $\nu_{0}(\mathcal{D})$ be the circuit packing number of $\mathcal{D}$, namely, the maximum number of vertex-disjoint circuits in $\mathcal{D}$. We establish below an upper bound on minranks of digraphs. This bound was also obtained independently by Chaudhry et al. [14, 15].

Proposition 5.2.3 (Circuit-packing bound). The following holds for every digraph $\mathcal{D}$ or order $n$ :

$$
\operatorname{minrk}_{q}(\mathcal{D}) \leq n-\nu_{0}(\mathcal{D})
$$

Proof. Suppose $\mathcal{D}$ contains $\nu_{0}(\mathcal{D})$ vertex-disjoint circuits $\mathcal{C}_{1}, \mathcal{C}_{2}, \ldots, \mathcal{C}_{\nu_{0}(\mathcal{D})}$, where

$$
\mathcal{C}_{i}=\left(u_{i, 1}, u_{i, 2}, \ldots, u_{i, n_{i}}\right), i \in\left[\nu_{0}(\mathcal{D})\right] .
$$

We construct a matrix $\boldsymbol{M}$ fitting $\mathcal{D}$ as follows. Let

$$
\mathcal{A} \triangleq[n] \backslash \cup_{i \in\left[\nu_{0}(\mathcal{D})\right]} \mathcal{V}\left(\mathcal{C}_{i}\right) .
$$

For $v \in \mathcal{A}$ let $\boldsymbol{M}_{v}=\boldsymbol{e}_{v}$. For $i \in\left[\nu_{0}(\mathcal{D})\right]$ and $s \in\left[n_{i}-1\right]$, let

$$
\boldsymbol{M}_{u_{i, s}}=\boldsymbol{e}_{u_{i, s}}-\boldsymbol{e}_{u_{i, s+1}},
$$

and let

$$
\boldsymbol{M}_{u_{i, n_{i}}}=\boldsymbol{e}_{u_{i, 1}}-\boldsymbol{e}_{u_{i, n_{i}}} .
$$

Clearly, $\boldsymbol{M}$ fits $\mathcal{D}$. Moreover, as

$$
\boldsymbol{M}_{u_{i, n_{i}}}=\sum_{s=1}^{n_{i}-1} \boldsymbol{M}_{u_{i, s}},
$$

we have

$$
\operatorname{rank}_{q}\left(\boldsymbol{M}_{\mathcal{V}\left(\mathcal{C}_{i}\right)}\right)=n_{i}-1
$$

for all $i \in\left[\nu_{0}(\mathcal{D})\right]$. Since $\mathcal{V}\left(\mathcal{C}_{i}\right)$ 's, $i \in\left[\nu_{0}(\mathcal{D})\right]$, are pairwise disjoint, we have

$$
\begin{aligned}
\operatorname{rank}_{q}(\boldsymbol{M}) & =\sum_{i=1}^{\nu_{0}(\mathcal{D})} \operatorname{rank}_{q}\left(\boldsymbol{M}_{\mathcal{V}\left(\mathcal{C}_{i}\right)}\right)+\operatorname{rank}_{q}\left(\boldsymbol{M}_{\mathcal{A}}\right) \\
& =\sum_{i=1}^{\nu_{0}(\mathcal{D})}\left(n_{i}-1\right)+\left(n-\sum_{i=1}^{\nu_{0}(\mathcal{D})} n_{i}\right) \\
& =n-\nu_{0}(\mathcal{D}) .
\end{aligned}
$$

Thus, $\operatorname{minrk}_{q}(\mathcal{D}) \leq n-\nu_{0}(\mathcal{D})$.

The clique-cover bound for graphs (see Theorem 3.2.11) can be easily extended to a bound for digraphs as follows.

Proposition 5.2.4. It holds for any digraph $\mathcal{D}$ that $\operatorname{minrk}_{q}(\mathcal{D}) \leq \operatorname{cc}(\mathcal{D})$.

Proof. Let $\mathcal{V}(\mathcal{D})=[n]$ and $\mathcal{W}=\left\{\mathcal{W}_{1}, \mathcal{W}_{2}, \ldots, \mathcal{W}_{\mathrm{cc}(\mathcal{D})}\right\}$ a minimum clique cover of


Fig. 5.4: Example where the circuit-packing bound outperforms the clique-cover bound
$\mathcal{D}$. Then $[n]=\cup_{i=1}^{\mathrm{cc}(\mathcal{D})} \mathcal{W}_{i}$. We aim to construct a matrix $\boldsymbol{M}$ that fits $\mathcal{D}$ and has rank at most $\operatorname{cc}(\mathcal{D})$. Notice that by definition, $\mathcal{W}_{1}, \mathcal{W}_{2}, \ldots, \mathcal{W}_{\mathrm{cc}(\mathcal{D})}$ partition $\mathcal{V}(\mathcal{D})$. For each $i \in[\operatorname{cc}(\mathcal{D})]$ and $j \in \mathcal{W}_{i}$, let $\boldsymbol{M}_{j}=\sum_{s \in \mathcal{W}_{i}} \boldsymbol{e}_{s}$. Then $\boldsymbol{M}$ fits $\mathcal{D}$ and

$$
\operatorname{rank}_{q}(\boldsymbol{M}) \leq \sum_{i=1}^{\mathrm{cc}(\mathcal{D})} \operatorname{rank}_{q}\left(\boldsymbol{M}_{\mathcal{W}_{i}}\right) \leq \sum_{i=1}^{\mathrm{cc}(\mathcal{D})} 1=\operatorname{cc}(\mathcal{D})
$$

Whereas for graphs the clique-cover bound is the best known bound, for digraphs that are not symmetric, this is not the case. The worst scenario for the clique-cover bound is when the digraph has no two arcs of opposite directions. For such a digraph, this bound becomes trivial, as the size of the smallest clique cover is equal to the order of the digraph. The following example emphasizes the fact that for certain digraphs, the circuit-packing bound can be significantly better than the clique-cover bound.

Example 5.2.5. Let $\mathcal{D}_{k}$ be the digraph of order $n=3 k$ depicted in Fig. 5.4. As there are no arcs of opposite directions, all cliques in $\mathcal{D}_{k}$ are of cardinality one. Therefore, the clique-cover bound gives $\operatorname{minrk}_{q}\left(\mathcal{D}_{k}\right) \leq 3 k$. On the other hand, as $\mathcal{D}_{k}$ contains $k$ vertex-disjoint circuits, namely $\mathcal{C}_{i}=(3 i+1,3 i+2,3 i+3)$ for $i=0,1, \ldots, k-1$, the circuit-packing bound yields $\operatorname{minrk}_{q}\left(\mathcal{D}_{k}\right) \leq 2 k=3 k-k$. The gap between the two bounds is a third of the order of the digraph.

### 5.2.2 Digraphs Attaining Circuit-Packing Bound

A feedback vertex (arc, respectively) set of $\mathcal{D}$ is a set of vertices (arcs, respectively) whose removal destroys all circuits in $\mathcal{D}$. Let $\tau_{0}(\mathcal{D})\left(\tau_{1}(\mathcal{D})\right.$, respectively) denote the minimum size of a feedback vertex (arc, respectively) set of $\mathcal{D}$. Then it is clear that $\alpha(\mathcal{D})=n-\tau_{0}(\mathcal{D})$.

Corollary 5.2.6. If $\nu_{0}(\mathcal{D})=\tau_{0}(\mathcal{D})$ then

$$
\operatorname{minrk}_{q}(\mathcal{D})=n-\nu_{0}(\mathcal{D})=n-\tau_{0}(\mathcal{D}) .
$$

Proof. By Corollary 5.2.2 and Proposition 5.2.3 we have

$$
n-\tau_{0}(\mathcal{D}) \leq \operatorname{minrk}_{q}(\mathcal{D}) \leq n-\nu_{0}(\mathcal{D}) .
$$

Hence, the proof follows.

When $\mathcal{D}$ satisfies $\nu_{0}(\mathcal{D})=\tau_{0}(\mathcal{D})$, we say that $\mathcal{D}$ satisfies the min-max vertex equality. In that case, the circuit-packing bound is actually tight. Similarly, let $\nu_{1}(\mathcal{D})$ denotes the maximum number of arc-disjoint circuits in $\mathcal{D}$. We say that $\mathcal{D}$ satisfies the min-max arc equality if $\nu_{1}(\mathcal{D})=\tau_{1}(\mathcal{D})$.

The first family of digraphs for which the circuit-packing bound is tight is the family of fully reducible flow digraphs [30]. A flow digraph is a digraph that contains a special vertex called root, from which any vertex is reachable by a path. A fully reducible flow digraph is a flow digraph that satisfies the property that every circuit $\mathcal{C}$ in the digraph has a unique vertex $v_{\mathcal{C}}$ such that every path from the root to a vertex of $\mathcal{C}$ must contain $v_{\mathcal{C}}$. Interestingly, it was proved by Shamir [56] that there is a linear time algorithm to find $\nu_{0}(\mathcal{D})=\tau_{0}(\mathcal{D})$ for a fully reducible flow digraph
$\mathcal{D}$. As a consequence, the minrank of a fully reducible flow digraph (recognizable in polynomial time with respect to its size [63]) can be calculated in linear time with respect to its size.

The second family of digraphs that satisfy the min-max vertex equality is the family of connectively reducible digraphs [61]. This family actually generalizes both the family of fully reducible flow digraphs and the family of cyclically reducible digraphs [66]. A polynomial time algorithm was provided by Szwarcfiter [61] to recognize a member of this family and subsequently find a maximum set of vertex-disjoint circuits as well as a minimum feedback vertex set. Therefore, by Corollary 5.2.6,

$$
\operatorname{minrk}_{q}(\mathcal{D})=n-\nu_{0}(\mathcal{D})=n-\tau_{0}(\mathcal{D})
$$

for a connectively reducible digraph $D$ and moreover, it can be found in polynomial time.

The third family of digraphs for which the circuit-packing bound is tight is the family of digraphs that pack [32]. A digraph packs if the min-max vertex equality holds for all of its subgraphs. The digraphs in this family are exactly ones that have no minor isomorphic to an odd double circuit or $F_{7}$, a special digraph of order 7 (interested readers may refer to [32] for more details, also for a structural characterization of this family of digraphs). For instance, strongly planar digraphs [32] belong to this family. As far as we know, there are no known polynomial time algorithms to find a minimum feedback vertex set of a digraph that packs.

The last three families of digraphs for which the circuit-packing bound is tight are families of line digraphs of planar digraphs, of fully reducible flow digraphs, and of (special) Eulerian digraphs.

Definition 5.2.7. Let $\mathcal{D}=(\mathcal{V}(\mathcal{D}), \mathcal{E}(\mathcal{D}))$ be a digraph. Then the digraph $\mathcal{L}=$
$(\mathcal{V}(\mathcal{L}), \mathcal{E}(\mathcal{L}))$ with $\mathcal{V}(\mathcal{L})=\mathcal{E}(\mathcal{D})$ and

$$
\mathcal{E}(\mathcal{L})=\left\{\left(e, e^{\prime}\right): e=(u, v) \in \mathcal{E}(\mathcal{D}), e^{\prime}=(v, w) \in \mathcal{E}(\mathcal{D})\right\}
$$

is called the line digraph of $\mathcal{D}$. We denote the line digraph of $\mathcal{D}$ by $\mathcal{L}(\mathcal{D})$. The digraph $\mathcal{D}$ is called a root digraph of $\mathcal{L}(\mathcal{D})$.

Lemma 5.2.8. $\nu_{0}(\mathcal{L}(\mathcal{D}))=\nu_{1}(\mathcal{D})$.

## Proof.

1. $\nu_{0}(\mathcal{L}(\mathcal{D})) \geq \nu_{1}(\mathcal{D})$. It suffices to show that the existence of a set of arc-disjoint circuits in $\mathcal{D}$ implies the existence of a set of vertex-disjoint circuits of the same size in $\mathcal{L}(\mathcal{D})$. Let $\left\{\mathcal{C}_{1}, \mathcal{C}_{2}, \ldots, \mathcal{C}_{k}\right\}$ be a set of arc-disjoint circuits in $\mathcal{D}$, where $\mathcal{C}_{i}=\left(v_{i, 1}, v_{i, 2}, \ldots, v_{i, r_{i}}\right), r_{i} \geq 2, i \in[k]$. Let $e_{i, j}=\left(v_{i, j}, v_{i, j+1}\right)$, for $i \in[k]$ and $j \in\left[r_{i}-1\right]$. Moreover, let $e_{i, r_{i}}=\left(v_{i, r_{i}}, v_{i, 1}\right)$ for $i \in[k]$. Let $\mathcal{C}_{i}^{\prime}=\left(e_{i, 1}, e_{i, 2}, \ldots, e_{i, r_{i}}\right)$ for $i \in[k]$. Then $\mathcal{C}_{i}^{\prime}$ is also a circuit in $\mathcal{L}(\mathcal{D})$ for every $i \in[k]$. Moreover, as the circuits $\mathcal{C}_{1}, \mathcal{C}_{2}, \ldots, \mathcal{C}_{k}$ share no common edges in $\mathcal{D}$, we deduce that $\mathcal{C}_{1}^{\prime}, \mathcal{C}_{2}^{\prime}, \ldots, \mathcal{C}_{k}^{\prime}$ share no common vertices in $\mathcal{L}(\mathcal{D})$. Therefore, they form a set of $k$ vertex-disjoint circuits in $\mathcal{L}(\mathcal{D})$.
2. $\nu_{0}(\mathcal{L}(\mathcal{D})) \leq \nu_{1}(\mathcal{D})$. It suffices to show that the existence of a set of vertexdisjoint circuits in $\mathcal{L}(\mathcal{D})$ implies the existence of a set of arc-disjoint circuits of the same size in $\mathcal{D}$. Let $\left\{\mathcal{C}_{1}^{\prime}, \mathcal{C}_{2}^{\prime}, \ldots, \mathcal{C}_{k}^{\prime}\right\}$ be a set of vertex-disjoint circuits in $\mathcal{L}(\mathcal{D})$, where $\mathcal{C}_{i}^{\prime}=\left\{e_{i, 1}, e_{i, 2}, \ldots, e_{i, r_{i}}\right\}$ for $i \in[k]$. Suppose that $e_{i, j}=$ $\left(v_{i, j}, v_{i, j+1}\right) \in \mathcal{E}(\mathcal{D})$ for $i \in[k]$ and $j \in\left[r_{i}\right]$, where $v_{i, j}$ and $v_{i, j+1}$ are vertices of $\mathcal{D}$. Then $v_{i, r_{i}+1} \equiv v_{i, 1}$ for $i \in[k]$. For each $i \in[k]$, consider the sequence of
(possibly repeated) vertices

$$
v_{i, 1}, v_{i, 2}, \ldots, v_{i, r_{i}+1}
$$

Since $v_{i, 1} \equiv v_{i, r_{i}+1}$ and $\left(v_{i, j}, v_{i, j+1}\right) \in \mathcal{E}(\mathcal{D})$ for all $j \in\left[r_{i}\right]$, there exist $j_{0}$ and $j_{1}$ such that
(a) $1 \leq j_{0}<j_{1} \leq r_{i}$;
(b) $v_{i, j_{0}} \equiv v_{i, j_{1}+1}$;
(c) $v_{i, j_{0}}, v_{i, j_{0}+1}, \ldots, v_{i, j_{1}}$ are distinct.

Then $\mathcal{C}_{i}=\left(v_{i, j_{0}}, v_{i, j_{0}+1}, \ldots, v_{i, j_{1}}\right)$ is a circuit in $\mathcal{D}$. Since the circuits $\mathcal{C}_{1}^{\prime}, \mathcal{C}_{2}^{\prime}, \ldots, \mathcal{C}_{k}^{\prime}$ share no common vertices in $\mathcal{L}(\mathcal{D})$, we obtain that the circuits $\mathcal{C}_{1}, \mathcal{C}_{2}, \ldots, \mathcal{C}_{k}$ share no common edges in $\mathcal{D}$.

Lemma 5.2.9. $\tau_{0}(\mathcal{L}(\mathcal{D}))=\tau_{1}(\mathcal{D})$.

Proof. Let $F=\left\{e_{1}, e_{2}, \ldots, e_{k}\right\}$, where $e_{i} \in \mathcal{E}(\mathcal{D})$ for $i \in[k]$, be an arbitrary set of arcs of $\mathcal{D}$. We can also consider $F$ as a set of vertices of $\mathcal{L}(\mathcal{D})$. It suffices to show that $F$ is a feedback arc set of $\mathcal{D}$ if and only if $F$ is a feedback vertex set of $\mathcal{L}(\mathcal{D})$, for every such set $F$.

Let $\mathcal{D}-F$ be the digraph obtained from $\mathcal{D}$ by removing all arcs in $F$. Let $\mathcal{L}(\mathcal{D})-F$ be the digraph obtained from $\mathcal{L}(\mathcal{D})$ by removing all vertices in $F$. Then $\mathcal{L}(\mathcal{D})-F=\mathcal{L}(\mathcal{D}-F)$. As shown in the proof of Lemma 5.2 .8 , the existence of a circuit in $\mathcal{D}-F$ would result in the existence of a circuit in $\mathcal{L}(\mathcal{D}-F)$ and vice versa. Therefore, $\mathcal{D}-F$ is acyclic if and only if $\mathcal{L}(\mathcal{D})-F$ is acyclic. Thus, $F$ is a feedback arc set of $\mathcal{D}$ if and only if $F$ is a feedback vertex set of $\mathcal{L}(\mathcal{D})$.

Proposition 5.2.10. Let $\mathcal{D}$ be a digraph. If $\nu_{1}(\mathcal{D})=\tau_{1}(\mathcal{D})$ then $\nu_{0}(\mathcal{L}(\mathcal{D}))=$ $\tau_{0}(\mathcal{L}(\mathcal{D}))$ and

$$
\operatorname{minrk}_{q}(\mathcal{L}(\mathcal{D}))=|\mathcal{V}(\mathcal{L}(\mathcal{D}))|-\nu_{0}(\mathcal{L}(\mathcal{D}))=|\mathcal{E}(\mathcal{D})|-\nu_{1}(\mathcal{D})
$$

Proof. Suppose that $\nu_{1}(\mathcal{D})=\tau_{1}(\mathcal{D})$. By Lemma 5.2.8 and Lemma 5.2.9, $\nu_{0}(\mathcal{L}(\mathcal{D}))=$ $\tau_{0}(\mathcal{L}(\mathcal{D}))$. Therefore, applying Corollary 5.2.6 to $\mathcal{L}(\mathcal{D})$ we obtain

$$
\operatorname{minrk}_{q}(\mathcal{L}(\mathcal{D}))=|\mathcal{V}(\mathcal{L}(\mathcal{D}))|-\nu_{0}(\mathcal{L}(\mathcal{D}))=|\mathcal{E}(\mathcal{D})|-\nu_{1}(\mathcal{D})
$$

Definition 5.2.11. A (di)graph that can be drawn on a plane in such a way that its (arcs) edges intersect only at their endpoints is called planar.

It is known that the min-max arc equality is satisfied for planar digraphs [48], for fully reducible flow digraphs [54], and for a special family of Eulerian digraphs [55]. Therefore, by Proposition 5.2.10, the min-max vertex equality is satisfied for the line digraphs of the members of these families. In summary, we have the following.

Corollary 5.2.12. The circuit-packing bound is tight for the following families of digraphs: connectively reducible digraphs, digraphs that pack, line digraphs of planar digraphs, line digraphs of fully reducible flow digraphs, and line digraphs of special Eulerian digraphs.

Definition 5.2.13. A digraph is called partially planar if all of its strongly connected components are planar.

Since the strongly connected components of a planar digraph are also planar, a planar digraph is partially planar. However, the converse is not always true, as
shown in Fig. 5.5.


Fig. 5.5: A partially planar digraph that is not planar

Proposition 5.2.14. There is a polynomial time algorithm to recognize the line digraph of a partially planar digraph and subsequently determine its minrank.

Proof.

## 1. Recognition Phase:

There is a one-to-one correspondence between the set of strongly connected components of order at least two of $\mathcal{D}$ and the set of strongly connected components of $\mathcal{L}(\mathcal{D})$ in the following sense. If $\mathcal{D}_{i}$ 's, $i \in[k]$, are all strongly connected components of $\mathcal{D}$ each of which contains at least two vertices, then $\mathcal{L}\left(\mathcal{D}_{i}\right)$ 's, $i \in[k]$, are all strongly connected components of $\mathcal{L}(\mathcal{D})$. Therefore, to determine whether a given digraph $\mathcal{L}$ is the line digraph of a partially planar digraph, it suffices to determine whether each of its strongly connected components $\mathcal{L}_{i}(i \in[k])$ is the line digraph of a planar digraph. Note also that we can find all strongly connected components of a digraph in linear time [62].

For each $i \in[k]$, employing a polynomial time algorithm, we can determine
whether $\mathcal{L}_{i}$ is a line digraph of a digraph [60]. If the answer is YES, then the algorithm also outputs a digraph $\mathcal{D}_{i}^{\prime}$, which is a root digraph of $\mathcal{L}_{i}$ and is strongly connected.

Suppose $\mathcal{L}=\mathcal{L}(\mathcal{D})$, where $\mathcal{D}$ is a digraph. Moreover, let $\mathcal{L}_{i}=\mathcal{L}\left(\mathcal{D}_{i}\right)$, where $\mathcal{D}_{i}$ 's, $i \in[k]$, are all strongly connected components of $\mathcal{D}$ of order at least two. By [35, Theorem 3], $\mathcal{D}_{i}^{\prime}$ and $\mathcal{D}_{i}$ are actually isomorphic, $i \in[k]$. Hence, to complete the Recognition Phase, one needs to test the planarity of $\mathcal{D}_{i}^{\prime}$ for every $i \in[k]$. It is well known that this task can be done in linear time [38]. Thus, the Recognition Phase can be done in polynomial time.

## 2. MinRank Computation Phase:

Upon the completion of the Recognition Phase, if it is confirmed that $\mathcal{L}$ is indeed the line digraph of a partially planar digraph, then the second phase is carried out to compute $\operatorname{minrk}_{q}(\mathcal{L})$. We show that this phase can also be done in polynomial time. Indeed, by Lemma 4.1.2, it suffices to show that $\operatorname{minrk}_{q}\left(\mathcal{L}_{i}\right)$ for $i \in[k]$ can be found in polynomial time.

On the one hand, since $\mathcal{D}_{i}^{\prime}$ (which is isomorphic to $\mathcal{D}_{i}$ ) is planar, as shown by Lucchesi and Younger [48], $\nu_{1}\left(\mathcal{D}_{i}^{\prime}\right)=\tau_{1}\left(\mathcal{D}_{i}^{\prime}\right)$. Therefore, by Proposition 5.2.10,

$$
\operatorname{minrk}_{q}\left(\mathcal{L}_{i}\right)=\left|\mathcal{V}\left(\mathcal{L}_{i}\right)\right|-\nu_{0}\left(\mathcal{L}_{i}\right)=\left|\mathcal{E}\left(\mathcal{D}_{i}^{\prime}\right)\right|-\nu_{1}\left(\mathcal{D}_{i}^{\prime}\right)
$$

On the other hand, $\nu_{1}\left(\mathcal{D}_{i}^{\prime}\right)$ can be computed in polynomial time [47]. Therefore $\operatorname{minrk}_{q}\left(\mathcal{L}_{i}\right)$ for each $i \in[k]$ can be computed in polynomial time. Thus, $\operatorname{minrk}_{q}(\mathcal{L})$ can be found in polynomial time.

In summary, we have the following.

Corollary 5.2.15. There are polynomial time algorithms to recognize a member and subsequently determine the minrank of that member of the following families of digraphs: connectively reducible digraphs (which includes fully reducible flow digraphs and cyclically reducible digraphs), and line digraphs of partially planar digraphs.

Note that since a strongly connected component of a fully reducible flow digraph may no longer be a fully reducible flow digraph, the arguments in the proof of Proposition 5.2.14 do not work for line digraphs of fully reducible flow digraphs.

We now discuss vector index codes and their transmission rates. Consider an ICSI instance $(n, n, \mathcal{X}, f)$. We treat each message $x_{i}(i \in[n])$ as a vector in $\mathbb{F}_{q}^{t}$ for some $t \geq 1$. An encoding function $\mathfrak{E}: \mathbb{F}_{q}^{t n} \rightarrow \mathbb{F}_{q}^{k}$ that enables each receiver $R_{i}$ to recover $x_{i}$ based on $\mathfrak{E}(\boldsymbol{x})$ and its side information is called a vector index code of block length $t$. We call the ratio $k / t$ the transmission rate of the vector index code (see $[2,28]$ ). An interesting task is to find an optimal vector index code, that is, a vector index code that achieves minimum transmission rate. When the block length $t$ equals one, the vector index codes become the scalar index codes, and the aforementioned problem becomes the familiar problem of finding a scalar index code of minimum length.

Consider an ICSI instance described by a digraph $\mathcal{D}$. In the remainder of this section, we show that if $\operatorname{minrk}_{q}(\mathcal{D})=\alpha(\mathcal{D})$ then an optimal scalar linear index code (block length $t=1$ ) always achieves the minimum transmission rate. As a consequence, we obtain several families of side information digraphs, listed in Corollary 5.2.12, for ICSI instances described by which, scalar linear index codes are already optimal. Before this work, only perfect graphs and acyclic digraphs are known to have this property.

Proposition 5.2.16. If $\operatorname{minrk}_{q}(\mathcal{D})=\alpha(\mathcal{D})$ then the optimal transmission rate of an

ICSI instance described by $\mathcal{D}$ is $\alpha(\mathcal{D})$. As a result, scalar linear index codes achieve the optimal rate.

Proof. First, observe that a vector index code for an ICSI instance described by $\mathcal{D}$ with block length $t \geq 1$ is a scalar index code for another ICSI instance described by $\mathcal{D}_{t}$ and vice versa, where $\mathcal{D}_{t}=\left(\mathcal{V}\left(\mathcal{D}_{t}\right), \mathcal{E}\left(\mathcal{D}_{t}\right)\right)$ is defined as follows. The vertex set is $\mathcal{V}\left(\mathcal{D}_{t}\right)=\{(i, r): i \in[n], r \in[t]\}$, and the arc set is

$$
\mathcal{E}\left(\mathcal{D}_{t}\right)=\{((i, r),(j, s)):(i, j) \in \mathcal{E}(\mathcal{D}), r, s \in[t]\}
$$

Note that in the ICSI instance described by $\mathcal{D}_{t}$, there are $n t$ receivers, each requests precisely one bit of information. There are now $n$ groups of receivers, in each of which all $t$ receivers own the same side information. Each of these groups of $t$ receivers corresponds to one receiver in the original instance.

Suppose that $\mathcal{I}=\left\{i_{1}, i_{2}, \ldots, i_{\alpha(\mathcal{D})}\right\}$ induces a maximum acyclic subgraph of $\mathcal{D}$. By the definition of $\mathcal{D}_{t}$, the set

$$
\mathcal{I}_{t}=\left\{\left(i_{j}, r\right): j \in[\alpha(\mathcal{D})], r \in[t]\right\}
$$

induces an acyclic subgraph of $\mathcal{D}_{t}$. Therefore

$$
\alpha\left(\mathcal{D}_{t}\right) \geq\left|\mathcal{I}_{t}\right|=|t \alpha(\mathcal{D})| .
$$

Hence, by [3, Theorem 6], any scalar index code for the ICSI instance described by $\mathcal{D}_{t}$ has length at least $\operatorname{t\alpha }(\mathcal{D})$. Therefore, any vector index code for the ICSI instance described by $\mathcal{D}$ has transmission rate at least $\alpha(\mathcal{D})=(t \alpha(\mathcal{D})) / t$, which is equal to the rate of an optimal scalar linear index code. We recall that a straightforward generalization of [4, Theorem 5] shows that $\operatorname{minrk}_{q}(\mathcal{D})(=\alpha(\mathcal{D})$ in this case) equals
the rate of an optimal scalar linear index code over $\mathbb{F}_{q}$. The proof is complete.

### 5.3 On MinRanks of Graphs Having Tree Structures of Type I

We present in this section a new family of graphs whose minranks can be found in polynomial time.

### 5.3.1 Tree Structure of Type I

Let $\mathscr{P}$ be a collection of finitely many families of graphs that satisfy the following requirements:

1. Each family is closed under the operation of taking induced subgraphs, that is, every induced subgraph of a member of a family of $\mathscr{P}$ also belongs to that family;
2. There is a polynomial time algorithm to recognize a member of each family;
3. There is a polynomial time algorithm to find the minrank of every member of each family.

For instance, we may choose $\mathscr{P}$ to be the collection of the following three families: perfect graphs $[3,19]$, outerplanar graphs $[8,68]$, and graphs of orders bounded by a constant. Instead of saying that a graph $\mathcal{G}$ belongs to a family in $\mathscr{P}$, with a slight abuse of notation, we often simply say that $\mathcal{G} \in \mathscr{P}$.

Let $U$ and $V$ be two disjoint nonempty sets of vertices of $\mathcal{G}$. Let

$$
\mathbf{s}_{\mathcal{G}}(U, V)=|\{\{u, v\}: u \in U, v \in V,\{u, v\} \in \mathcal{E}(\mathcal{G})\}|,
$$

denotes the number of edges each of which has one endpoint in $U$ and the other endpoint in $V$.

Definition 5.3.1. A connected graph $\mathcal{G}=(\mathcal{V}(\mathcal{G}), \mathcal{E}(\mathcal{G}))$ is said to have a tree structure of Type $I$ if there exists a partition $\Gamma=\left[V_{1}, V_{2}, \ldots, V_{k}\right]$ of the vertex set $\mathcal{V}(\mathcal{G})$ that satisfies the following three requirements:
(R1) The $V_{i}$-induced subgraph $\mathcal{G}_{i}$ of $\mathcal{G}$ belongs to a family in $\mathscr{P}$, for every $i \in[k]$;
$(\mathrm{R} 2) \mathrm{s}_{\mathcal{G}}\left(V_{i}, V_{j}\right) \in\{0,1\}$ for every $i \neq j$;
(R3) The graph $T=(\mathcal{V}(T), \mathcal{E}(T))$, where $\mathcal{V}(T)=[k]$ and

$$
\mathcal{E}(T)=\left\{\{i, j\}: \mathbf{s}_{\mathcal{G}}\left(V_{i}, V_{j}\right)=1\right\}
$$

is a rooted tree; The tree $T$ can also be thought of as a graph obtained from $\mathcal{G}$ by contracting each $\mathcal{V}_{i}$ to a single vertex.

The 2-tuple $\mathscr{T}=(\Gamma, T)$ is called a tree structure of Type $I$ of $\mathcal{G}$.


Fig. 5.6: A tree structure of Type I of a graph $\mathcal{G}$

If a tree structure of Type I $\mathscr{T}=(\Gamma, T)$ of $\mathcal{G}$ is given, where $\Gamma=\left[V_{1}, V_{2}, \ldots, V_{k}\right]$, then we can define the following terms:

1. Each $V_{i}$-induced subgraph $\mathcal{G}_{i}$ of $\mathcal{G}$ is called a node of $\mathscr{T}$;
2. If $i$ is the parent of $j$ in $T$, then $G_{i}$ is called the parent (node) of $\mathcal{G}_{j}$ in $\mathscr{T}$; We also refer to $\mathcal{G}_{j}$ as a child (node) of $\mathcal{G}_{i}$; Nodes with no children are called leaves; The node with no parent is called the root;
3. For each $i \in[k]$ let $\mathcal{S}_{i}$ be the subgraph of $\mathcal{G}$ induced by $V_{i} \cup\left(\cup_{j \in \operatorname{des}_{T}(i)} V_{j}\right)$, where $\operatorname{des}_{T}(i)$ denotes the set of descendants of $i$ in $T$;
4. If $\mathcal{G}_{j}$ is a child of $\mathcal{G}_{i}$, and $\{u, v\} \in \mathcal{E}(\mathcal{G})$, where $u \in V_{i}$ and $v \in V_{j}$, then $u$ is called a downward connector ( DC ) of $\mathcal{G}_{i}$ and $v$ is called the upward connector (UC) of $\mathcal{G}_{j}$; Each node may have several DCs but at most one UC; DCs and UC of a node are called connectors of that node.
5. Let $\operatorname{mdc}(\mathscr{T})$ denote the maximum number of DCs of a node of $\mathscr{T}$.

For any $c>0$ we define the following two families of connected graphs

$$
\begin{aligned}
& \mathscr{F}_{1}(c) \triangleq\{\mathcal{G}: \mathcal{G} \text { is connected and has a tree structure } \mathscr{T}=(\Gamma, T) \text { of Type I } \\
& \text { satisfying } \operatorname{mdc}(\mathscr{T}) \leq c\}
\end{aligned}
$$

and

$$
\begin{aligned}
\mathscr{F}_{2}(c) \triangleq\{\mathcal{G}: & \mathcal{G} \text { is connected and has a tree structure } \mathscr{T}=(\Gamma, T) \text { of Type I } \\
& \text { satisfying } \operatorname{mdc}(\mathscr{T}) \leq c \text { and each node of } \mathscr{T} \text { is } 2 \text {-edge connected }\},
\end{aligned}
$$

A tree structure of Type I of a graph $\mathcal{G}$ that proves the membership of $\mathcal{G}$ in $\mathscr{F}_{1}(c)$ or $\mathscr{F}_{2}(c)$ is called a relevant tree structure of $\mathcal{G}$.

### 5.3.2 An Algorithm for MinRanks of Graphs in $\mathscr{F}_{1}(c)$

In this section we establish that the minrank of a member of $\mathscr{F}_{1}(c)$ can be found in polynomial time.

Theorem 5.3.2. Let $c>0$ be a constant and $\mathcal{G} \in \mathscr{F}_{1}(c)$. Suppose further that a tree structure of Type $I \mathscr{T}=(\Gamma, T)$ of $\mathcal{G}$ with $\operatorname{mdc}(\mathscr{T}) \leq c$ is known. Then there is an algorithm that computes the minrank of $\mathcal{G}$ in polynomial time.

To prove Theorem 5.3.2, we describe below an algorithm that computes the minrank of $\mathcal{G}$ when $\mathcal{G} \in \mathscr{F}_{1}(c)$ and investigate its complexity.

First, we introduce some notation which is used throughout this section. If $v$ is any vertex of a graph $\mathcal{G}$, then $\mathcal{G}-v$ denotes the graph obtained from $\mathcal{G}$ by removing $v$ and all edges incident to $v$. In general, if $V$ is any set of vertices, then $\mathcal{G}-V$ denotes the graph obtained from $\mathcal{G}$ by removing all vertices in $V$ and all edges incident to any vertex in $V$. In other words, $\mathcal{G}-V$ is the subgraph of $\mathcal{G}$ induced by $\mathcal{V}(\mathcal{G}) \backslash V$. Note that if $\mathcal{G} \in \mathscr{P}$ then the minrank of $\mathcal{G}-V$ can be computed in polynomial time for every subset $V \subset \mathcal{V}(\mathcal{G})$. If $e=\{u, v\} \in \mathcal{E}(\mathcal{G})$ then $\mathcal{G}-e$ denotes the graph obtained from $\mathcal{G}$ by removing the edge $e$ without removing its endpoints $u$ and $v$. In general, if $E$ is any set of edges, then $\mathcal{G}-E$ denotes the graph obtained from $\mathcal{G}$ by removing all edges in $E$ without removing their endpoints. The union of two or more graphs is a graph whose vertex set and edge set are the unions of the vertex sets and of the edge sets of the original graphs, respectively.

The following results from [8] are particularly useful in our discussion. Their proofs can be found in the full version of [8] at http://www.openu.ac.il/home/ mikel/papers/outer.pdf.

Lemma 5.3.3 ([8]). Let $v$ be a vertex of a graph $\mathcal{G}$. Then

$$
\operatorname{minrk}_{q}(\mathcal{G}-v) \leq \operatorname{minrk}_{q}(\mathcal{G}) \leq \operatorname{minrk}_{q}(\mathcal{G}-v)+1
$$

Lemma 5.3.4 ([8]). Let $\mathcal{G}_{1}$ and $\mathcal{G}_{2}$ be two graphs with one common vertex $v$. Then

$$
\begin{aligned}
\operatorname{minrk}_{q}\left(\mathcal{G}_{1} \cup \mathcal{G}_{2}\right)= & \operatorname{minrk}_{q}\left(\mathcal{G}_{1}-v\right)+\operatorname{minrk}_{q}\left(\mathcal{G}_{2}-v\right) \\
& +\left(\operatorname{minrk}_{q}\left(\mathcal{G}_{1}\right)-\operatorname{minrk}_{q}\left(\mathcal{G}_{1}-v\right)\right)\left(\operatorname{minrk}_{q}\left(\mathcal{G}_{2}\right)-\operatorname{minrk}_{q}\left(\mathcal{G}_{2}-v\right)\right) .
\end{aligned}
$$

In other words, the minrank of $\mathcal{G}_{1} \cup \mathcal{G}_{2}$ can be computed explicitly based on the minranks of $\mathcal{G}_{1}, \mathcal{G}_{1}-v, \mathcal{G}_{2}$, and $\mathcal{G}_{2}-v$.

## ALGO-1:

Suppose $\mathcal{G} \in \mathscr{F}_{1}(c)$ and a relevant tree structure of $\mathcal{G}$ is given. The algorithm computes the minrank by dynamic programming in a bottom-up manner, from the leaves of $\mathscr{T}$ to its root. Suppose that $\Gamma=\left[V_{1}, V_{2}, \ldots, V_{k}\right]$ and $\mathcal{G}_{i}$ is induced by $V_{i}$ for $i \in[k]$. Let $v_{i}$ be the UC (if any) of $\mathcal{G}_{i}$ for $i \in[k]$. For each $i$, ALGO-1 maintains a table which contains the two values, namely, minranks of $\mathcal{S}_{i}$ and $\mathcal{S}_{i}-v_{i}$. The minrank of the latter is omitted if $\mathcal{G}_{i}$ is the root node of $\mathscr{T}$. A key point is that the minranks of $\mathcal{S}_{i}$ and $\mathcal{S}_{i}-v_{i}$ can be computed in polynomial time from the minranks of $\mathcal{S}_{j}$ 's and of $\left(\mathcal{S}_{j}-v_{j}\right)$ 's where $\mathcal{G}_{j}$ 's are children of $\mathcal{G}_{i}$, and from the minranks of at most $2^{c}$ subgraphs of $\mathcal{G}_{i}$. Each of these subgraphs is obtained from $\mathcal{G}_{i}$ by removing a subset of a set that consists of at most $c$ vertices of $\mathcal{G}$. When the minrank of $\mathcal{S}_{i_{0}}$ is determined, where $G_{i_{0}}$ is the root of $\mathscr{T}$, the minrank of $\mathcal{G}$ is found.

## At the leaf-nodes:

Suppose $\mathcal{G}_{i}$ is a leaf and $v_{i}$ is its UC. Since $\mathcal{G}_{i}$ has no children, $\mathcal{S}_{i} \equiv \mathcal{G}_{i}$. Hence,

$$
\operatorname{minrk}_{q}\left(\mathcal{S}_{i}\right)=\operatorname{minrk}_{q}\left(\mathcal{G}_{i}\right)
$$

and

$$
\operatorname{minrk}_{q}\left(\mathcal{S}_{i}-v_{i}\right)=\operatorname{minrk}_{q}\left(\mathcal{G}_{i}-v_{i}\right)
$$

Both of these values can be computed in polynomial time, as $\mathcal{G}_{i} \in \mathscr{P}$.

## At the intermediate nodes:

Suppose the minranks of $\mathcal{S}_{j}$ and $\mathcal{S}_{j}-v_{j}$ are known for all $j$ such that $\mathcal{G}_{j}$ is a child of $\mathcal{G}_{i}$. The goal of the algorithm at this step is to compute the minranks of $\mathcal{S}_{i}$ and $\mathcal{S}_{i}-v_{i}$ in polynomial time. It is complicated to analyze directly the general case where $\mathcal{G}_{i}$ has an arbitrary number (at most $c$ ) of downward connectors. Therefore, we first consider a special case where $\mathcal{G}_{i}$ has only one downward connector (Case 1 ). The results established in this case are then used to investigate the general case (Case 2).

Case 1: $\mathcal{G}_{i}$ has only one DC $u$ and has $r$ children, namely $\mathcal{G}_{j_{1}}, \mathcal{G}_{j_{2}}, \ldots, \mathcal{G}_{j_{r}}$, all of which are connected to $\mathcal{G}_{i}$ via $u$.

Let $\mathcal{K}$ be the subgraph of $\mathcal{G}$ induced by the following set of vertices

$$
\mathcal{V}(\mathcal{K})=\mathcal{V}\left(\mathcal{S}_{j_{1}}\right) \cup \mathcal{V}\left(\mathcal{S}_{j_{2}}\right) \cup \cdots \cup \mathcal{V}\left(\mathcal{S}_{j_{r}}\right) \cup\{u\} .
$$

Notice that the graphs $\mathcal{G}_{i}$ and $\mathcal{K}$ share exactly one vertex, namely, $u$. Hence by Lemma 5.3.4, once the minranks of $\mathcal{G}_{i}, \mathcal{G}_{i}-u, \mathcal{K}$, and $\mathcal{K}-u$ are known, the minrank of $\mathcal{S}_{i}=\mathcal{G}_{i} \cup \mathcal{K}$ can be explicitly computed. Similarly, if $v_{i} \neq u$ and the minranks of $\mathcal{G}_{i}-v_{i}, \mathcal{G}_{i}-v_{i}-u, \mathcal{K}$, and $\mathcal{K}-u$ are known, the minrank of $\mathcal{S}_{i}-v_{i}=\left(\mathcal{G}_{i}-v_{i}\right) \cup \mathcal{K}$ can be explicitly computed. Observe also that if $v_{i} \equiv u$


Fig. 5.7: $\mathcal{G}_{i}$ has only one downward connector
then by Lemma 4.1.1,

$$
\operatorname{minrk}_{q}\left(\mathcal{S}_{i}-v_{i}\right)=\operatorname{minrk}_{q}\left(\mathcal{G}_{i}-u\right)+\operatorname{minrk}_{q}(\mathcal{K}-u)
$$

Again by Lemma 4.1.1,

$$
\operatorname{minrk}_{q}(\mathcal{K}-u)=\sum_{\ell=1}^{r} \operatorname{minrk}_{q}\left(\mathcal{S}_{j_{\ell}}\right)
$$

which is known. Moreover, as $\mathcal{G}_{i} \in \mathscr{P}$, the minranks of $\mathcal{G}_{i}, \mathcal{G}_{i}-v_{i}, \mathcal{G}_{i}-u$, and $\mathcal{G}_{i}-v_{i}-u$ can be determined in polynomial time. Therefore it remains to compute the minrank of $\mathcal{K}$ efficiently. According to the following claim, the minrank of $\mathcal{K}$ can be explicitly computed based on the knowledge of the minranks of $\mathcal{S}_{j_{\ell}}$ and $\mathcal{S}_{j_{\ell}}-v_{j_{\ell}}$ for $\ell \in[r]$.

Claim 5.3.5. The minrank of $\mathcal{K}$ is equal to

$$
\begin{cases}\operatorname{minrk}_{q}(\mathcal{K}-u), & \text { if } \exists h \in[r] \text { s.t. } \operatorname{minrk}_{q}\left(\mathcal{S}_{j_{h}}-v_{j_{h}}\right)=\operatorname{minrk}_{q}\left(\mathcal{S}_{j_{h}}\right)-1, \\ \operatorname{minrk}_{q}(\mathcal{K}-u)+1, & \text { otherwise. }\end{cases}
$$

Proof. Suppose there exists $h \in[r]$ such that

$$
\operatorname{minrk}_{q}\left(\mathcal{S}_{j_{h}}-v_{j_{h}}\right)=\operatorname{minrk}_{q}\left(\mathcal{S}_{j_{h}}\right)-1
$$

By Lemma 5.3.3,

$$
\operatorname{minrk}_{q}(\mathcal{K}) \geq \operatorname{minrk}_{q}(\mathcal{K}-u)
$$

Therefore, in this case it suffices to show that a matrix that fits $\mathcal{K}$ and has rank equal to $\operatorname{minrk}_{q}(\mathcal{K}-u)$ exists. Indeed, such a matrix $\boldsymbol{M}$ can be constructed as follows:

1. Its sub-matrix restricted on $\mathcal{V}\left(\mathcal{S}_{j_{\ell}}\right)$ for $\ell \neq h$ has rank equal to $\operatorname{minrk}_{q}\left(\mathcal{S}_{j_{\ell}}\right)$;
2. Its sub-matrix restricted on $\mathcal{V}\left(\mathcal{S}_{j_{h}}\right) \backslash\left\{v_{j_{h}}\right\}$ has rank equal to $\operatorname{minrk}_{q}\left(\mathcal{S}_{j_{h}}-\right.$ $\left.v_{j_{h}}\right) ;$
3. $\boldsymbol{M}_{u}=\boldsymbol{M}_{v_{j_{h}}}=\boldsymbol{e}_{u}+\boldsymbol{e}_{v_{j_{h}}}$;
4. All other entries are zero.

Since the sets $\mathcal{V}\left(\mathcal{S}_{j_{\ell}}\right)(\ell \neq h), \mathcal{V}\left(\mathcal{S}_{j_{h}}\right) \backslash\left\{v_{j_{h}}\right\}$, and $\left\{u, v_{j_{h}}\right\}$ are pairwise disjoint, the above requirements can be met without any contradiction arising.

Moreover,

$$
\begin{aligned}
\operatorname{rank}_{q}(\boldsymbol{M}) & =\sum_{\ell \neq h} \operatorname{minrk}_{q}\left(\mathcal{S}_{j_{\ell}}\right)+\operatorname{minrk}_{q}\left(\mathcal{S}_{j_{h}}-v_{j_{h}}\right)+1 \\
& =\sum_{\ell \neq h} \operatorname{minrk}_{q}\left(\mathcal{S}_{j_{\ell}}\right)+\operatorname{minrk}_{q}\left(\mathcal{S}_{j_{h}}\right) \\
& =\operatorname{minrk}_{q}(\mathcal{K}-u) .
\end{aligned}
$$

We now suppose that $\operatorname{minrk}_{q}\left(\mathcal{S}_{j_{\ell}}-v_{j_{\ell}}\right)=\operatorname{minrk}_{q}\left(\mathcal{S}_{j_{\ell}}\right)$ for all $\ell \in[r]$. We prove that

$$
\operatorname{minrk}_{q}(\mathcal{K})=\operatorname{minrk}_{q}(\mathcal{K}-u)+1
$$

by induction on $r$.

1. The base case: $r=1$.


Fig. 5.8: The base case when $r=1$

Let $\mathcal{J}=(\mathcal{V}(\mathcal{J}), \mathcal{E}(\mathcal{J}))$ where $\mathcal{V}(\mathcal{J})=\left\{u, v_{j_{1}}\right\}$ and $\mathcal{E}(\mathcal{J})=\left\{\left\{u, v_{j_{1}}\right\}\right\}$.
Then $\mathcal{K}=\mathcal{J} \cup \mathcal{S}_{j_{1}}$ and $\mathcal{V}(\mathcal{J}) \cap \mathcal{V}\left(\mathcal{S}_{j_{1}}\right)=\left\{v_{j_{1}}\right\}$. Moreover,

$$
\operatorname{minrk}_{q}(\mathcal{J})=\operatorname{minrk}_{q}\left(\mathcal{J}-v_{j_{1}}\right)=1
$$

Therefore by Lemma 5.3.4,

$$
\begin{aligned}
\operatorname{minrk}_{q}(\mathcal{K}) & =\operatorname{minrk}_{q}\left(\mathcal{S}_{j_{1}}-v_{j_{1}}\right)+\operatorname{minrk}_{q}\left(\mathcal{J}-v_{j_{1}}\right) \\
& =\operatorname{minrk}_{q}\left(\mathcal{S}_{j_{1}}\right)+1 \\
& =\operatorname{minrk}_{q}(\mathcal{K}-u)+1
\end{aligned}
$$

2. The inductive step: suppose that the assertion holds for $r \geq 1$. We aim to show that it also holds for $r+1$.


Fig. 5.9: The inductive step

Let $\mathcal{J}$ be the subgraph of $\mathcal{G}$ induced by

$$
\{u\} \cup\left(\cup_{\ell=1}^{r} \mathcal{V}\left(\mathcal{S}_{j_{\ell}}\right)\right)
$$

Since $\operatorname{minrk}_{q}\left(\mathcal{S}_{j_{\ell}}-v_{j_{\ell}}\right)=\operatorname{minrk}_{q}\left(\mathcal{S}_{j_{\ell}}\right)$ for all $\ell \in[r]$, by the induction
hypothesis, we have

$$
\operatorname{minrk}_{q}(\mathcal{J})=\operatorname{minrk}_{q}(\mathcal{J}-u)+1
$$

Let $\mathcal{I}$ be the subgraph of $\mathcal{G}$ induced by $\{u\} \cup \mathcal{V}\left(\mathcal{S}_{j_{r+1}}\right)$. As

$$
\operatorname{minrk}_{q}\left(\mathcal{S}_{j_{r+1}}-v_{j_{r+1}}\right)=\operatorname{minrk}_{q}\left(\mathcal{S}_{j_{r+1}}\right),
$$

similar arguments as in the base case yield

$$
\operatorname{minrk}_{q}(\mathcal{I})=\operatorname{minrk}_{q}(\mathcal{I}-u)+1
$$

Applying Lemma 5.3.4 to the graphs $\mathcal{I}$ and $\mathcal{J}$ we obtain

$$
\begin{aligned}
\operatorname{minrk}_{q}(\mathcal{K}) & =\operatorname{minrk}_{q}(\mathcal{I} \cup \mathcal{J}) \\
& =\operatorname{minrk}_{q}(\mathcal{I}-u)+\operatorname{minrk}_{q}(\mathcal{J}-u)+1 \\
& =\sum_{\ell=1}^{r+1} \operatorname{minrk}_{q}\left(\mathcal{S}_{j_{\ell}}\right)+1,
\end{aligned}
$$

which is equal to $\operatorname{minrk}_{q}(\mathcal{K}-u)+1$.

Case 2: $\mathcal{G}_{i}$ has $d$ DCs, namely, $u_{1}, u_{2}, \ldots, u_{d}$. Let $\left\{\mathcal{G}_{j}: j \in I_{t}\right\}$ for $1 \leq t \leq d$, be the set of children of $\mathcal{G}_{i}$ connected to $\mathcal{G}_{i}$ via $u_{t}$.

Recall that the goal of the algorithm is to compute the minranks of $\mathcal{S}_{i}$ and $\mathcal{S}_{i}-v_{i}$ in polynomial time, given that the minranks of $\mathcal{S}_{j}$ and $\mathcal{S}_{j}-v_{j}$ are known for all children $\mathcal{G}_{j}$ 's of $\mathcal{G}_{i}$.

For each $t \in[d]$ let $\mathcal{K}_{t}$ be the subgraph of $\mathcal{G}$ induced by the following set of


Fig. 5.10: $\mathcal{G}_{i}$ has several downward connectors
vertices

$$
\left\{u_{t}\right\} \cup\left(\cup_{j \in I_{t}} \mathcal{V}\left(\mathcal{S}_{j}\right)\right)
$$

As in Case 1, based on the minranks of $\mathcal{S}_{j}$ 's and $\mathcal{S}_{j}-v_{j}$ for $j \in I_{t}$, it is possible to compute the minranks of $\mathcal{K}_{t}$ and $\mathcal{K}_{t}-u_{t}$ explicitly for all $t \in[d]$.

Let

$$
\mathcal{N}_{1}=\mathcal{K}_{1} \cup \mathcal{G}_{i}
$$

and

$$
\mathcal{N}_{t}=\mathcal{N}_{t-1} \cup \mathcal{K}_{t},
$$

for every $t \in[d]$ and $t \geq 2$. Observe that $\mathcal{N}_{d} \equiv \mathcal{S}_{i}$. Below we show how the algorithm computes the minranks of $\mathcal{N}_{d}$ and $\mathcal{N}_{d}-v_{i}$ inductively in polynomial time.

1. At the base case, the minranks of $\mathcal{N}_{1}-U$, for every subset $U \subseteq\left\{v_{i}, u_{2}, u_{3}\right.$,
$\left.\ldots, u_{d}\right\}$, are computed as follows.
If $v_{i} \equiv u_{1} \in U$, then

$$
\mathcal{N}_{1}-U=\left(\mathcal{K}_{1}-u_{1}\right) \cup\left(\mathcal{G}_{i}-U\right)
$$

Since

$$
\mathcal{V}\left(\mathcal{K}_{1}-u_{1}\right) \cup \mathcal{V}\left(\mathcal{G}_{i}-U\right)=\varnothing,
$$

by Lemma 4.1.1,

$$
\operatorname{minrk}_{q}\left(\mathcal{N}_{1}-U\right)=\operatorname{minrk}_{q}\left(\mathcal{K}_{1}-u_{1}\right)+\operatorname{minrk}_{q}\left(\mathcal{G}_{i}-U\right),
$$

which is computable in polynomial time.
Suppose that either $v_{i} \not \equiv u_{1}$ or $v_{i} \notin U$. By Lemma 5.3.4, since

$$
\mathcal{N}_{1}-U=\mathcal{K}_{1} \cup\left(\mathcal{G}_{i}-U\right)
$$

and

$$
\mathcal{V}\left(\mathcal{K}_{1}\right) \cap \mathcal{V}\left(\mathcal{G}_{i}-U\right)=\left\{u_{1}\right\}
$$

the minrank of $\mathcal{N}_{1}-U$ can be determined based on the minranks of $\mathcal{K}_{1}$, $\mathcal{K}_{1}-u_{1}, \mathcal{G}_{i}-U$, and $\mathcal{G}_{i}-U-u_{1}$. The minranks of these graphs are either known or computable in polynomial time. As $\operatorname{mdc}(\mathscr{T}) \leq c$, there are at most $2^{d} \leq 2^{c}$ (a constant) such subsets $U$. Hence, the total computation in the base case can be done in polynomial time.
2. At the inductive step, suppose that the minrank of $\mathcal{N}_{t-1}-U, t \geq 2$, for every subset $U \subseteq\left\{v_{i}, u_{t}, u_{t+1}, \ldots, u_{d}\right\}$ are known. Our goal is to show that the minrank of $\mathcal{N}_{t}-V$ for every subset $V \subseteq\left\{v_{i}, u_{t+1}, u_{t+2}, \ldots, u_{d}\right\}$
can be determined in polynomial time. Observe again that there are at most $2^{c}$ such subsets $V$.

If $v_{i} \equiv u_{t}$ and $v_{i} \in V$, then

$$
\mathcal{N}_{t}-V=\left(\mathcal{N}_{t-1}-V\right) \cup\left(\mathcal{K}_{t}-u_{t}\right) .
$$

Moreover, as we have

$$
\mathcal{V}\left(\mathcal{N}_{t-1}-V\right) \cap \mathcal{V}\left(\mathcal{K}_{t}-u_{t}\right)=\varnothing
$$

by Lemma 4.1.1,

$$
\operatorname{minrk}_{q}\left(\mathcal{N}_{t}-V\right)=\operatorname{minrk}_{q}\left(\mathcal{N}_{t-1}-V\right)+\operatorname{minrk}_{q}\left(\mathcal{K}_{t}-u_{t}\right)
$$

which is known.
Suppose that either $v_{i} \not \equiv u_{t}$ or $v_{i} \notin V$. Since

$$
\mathcal{N}_{t}-V=\left(\mathcal{N}_{t-1}-V\right) \cup \mathcal{K}_{t},
$$

and

$$
\mathcal{V}\left(\mathcal{N}_{t-1}-V\right) \cap \mathcal{V}\left(\mathcal{K}_{t}\right)=\left\{u_{t}\right\},
$$

the minrank of $\mathcal{N}_{t}-V$ can be computed based on the minranks of $\mathcal{N}_{t-1}-$ $V, \mathcal{N}_{t-1}-V-u_{t}, \mathcal{K}_{t}$, and $\mathcal{K}_{t}-u_{t}$, which are all available from the previous inductive step.

When the inductive process reaches $t=d$, the minranks of $\mathcal{N}_{d}$ and $\mathcal{N}_{d}-v_{i}$ are found, as desired. The analysis of Case 2 is completed.

When the algorithm reaches the root node of $\mathscr{T}$, the minrank of $\mathcal{G}$ is found.

### 5.3.3 An Algorithm to Recognize a Graph in $\mathscr{F}_{2}(c)$

In order for ALGO-1 to work, it is assumed that a relevant tree structure of the input graph $\mathcal{G} \in \mathscr{F}_{1}(c)$ is given. Therefore, the next question is how to design an algorithm that recognizes a graph in that family and subsequently finds a relevant tree structure for that graph in polynomial time. Unfortunately, we are unable to provide such an algorithm at the moment. However, the same task is possible for a sub-family of $\mathscr{F}_{1}(c)$, namely, $\mathscr{F}_{2}(c)$.

Theorem 5.3.6. Let $c>0$ be any constant. Then there is a polynomial time algorithm that recognizes a member of $\mathscr{F}_{2}(c)$. Moreover, this algorithm also outputs a relevant tree structure of that member.

In order to prove Theorem 5.3.6, we introduce ALGO-2 (Fig. 5.11). We show below that ALGO-2 indeed does what it is claimed to do.

Claim 5.3.7. If ALGO-2 terminates successfully, that is, it terminates without any error message printed out, then $\mathcal{G} \in \mathscr{F}_{2}(c)$ and the output is a relevant tree structure of $\mathcal{G}$.

Proof. Suppose that ALGO-2 terminates successfully. In the Splitting Phase, the algorithm first splits $\mathcal{G}$ into two vertex-disjoint components (subgraphs) that are connected to each other by exactly one edge in $\mathcal{G}$. It then keeps splitting the existing components, whenever possible, each into two new smaller (vertex-disjoint) components that are connected to each other by exactly one edge in the original

ALGO-2:
Input: A connected graph $\mathcal{G}=(\mathcal{V}(\mathcal{G}), \mathcal{E}(\mathcal{G}))$ and a constant $c>0$.
Output: If $\mathcal{G} \in \mathscr{F}_{2}(c)$, the algorithm prints out a confirmation message, namely " $\mathcal{G} \in \mathscr{F}_{2}(c)$ ", and then returns a relevant tree structure of $\mathcal{G}$. Otherwise, it prints out an error message " $\mathcal{G} \notin \mathscr{F}_{2}(c)$ ".
Initialization: Create two empty queues, $\mathcal{Q}_{1}$ and $\mathcal{Q}_{2}$, which contains graphs as their elements. Push $\mathcal{G}$ into $\mathcal{Q}_{1}$.

## Splitting Phase:

while $\mathcal{Q}_{1} \neq \varnothing$ do
for $\mathcal{A}=(\mathcal{V}(\mathcal{A}), \mathcal{E}(\mathcal{A})) \in \mathcal{Q}_{1}$ do
if there exist $U$ and $\mathcal{V}$ that partition $\mathcal{V}(\mathcal{A})$ and $\mathrm{s}_{\mathcal{A}}(U, V)=1$ then
Let $\mathcal{B}$ and $\mathcal{C}$ be subgraphs of $\mathcal{A}$ induced by $U$ and $V$, respectively
Push $B$ and $C$ into $\mathcal{Q}_{1}$
else if $\mathcal{A} \in \mathscr{P}$ then
Push $\mathcal{A}$ into $\mathcal{Q}_{2}$
else
Print the error message " $\mathcal{G} \notin \mathscr{F}_{2}(c)$ " and exit
end if
end for
end while

## Verifying Phase:

Suppose $\mathcal{Q}_{2}$ contains $k^{\prime}$ graphs $\mathcal{A}_{1}, \mathcal{A}_{2}, \ldots, \mathcal{A}_{k^{\prime}}$. The number of connectors of each graphs $\mathcal{A}_{i}$ for $i \in\left[k^{\prime}\right]$ is computed. If $\mathcal{A}_{i}$ has at most $c+1$ connectors for every $i$ then print out the message " $\mathcal{G} \in \mathscr{F}_{2}(c)$ " and output the vertex partition $\left[\mathcal{V}\left(\mathcal{A}_{1}\right), \ldots, \mathcal{V}\left(\mathcal{A}_{k^{\prime}}\right)\right]$ together with the rooted tree $T^{\prime}=\left(\mathcal{V}\left(T^{\prime}\right), \mathcal{E}\left(T^{\prime}\right)\right)$ constructed as follows. Let $\mathcal{V}\left(T^{\prime}\right)=\left[k^{\prime}\right]$ and $\mathcal{E}\left(T^{\prime}\right)=\left\{\{i, j\}: \mathrm{s}_{\mathcal{G}}\left(\mathcal{V}\left(\mathcal{A}_{i}\right), \mathcal{V}\left(\mathcal{A}_{j}\right)\right)=\right.$ $1\}$. The root of $T^{\prime}$ is $i_{0}$, where $A_{i_{0}}$ is any node that has at most $c$ connectors. Otherwise, print out the error message " $\mathcal{G} \notin \mathscr{F}_{2}(c)$ " and exit.

Fig. 5.11: ALGO-2


Fig. 5.12: Splitting Phase of ALGO-2
component (see Fig. 5.12). A straightforward inductive argument shows the following

1. Throughout the Splitting Phase, the vertex sets that induce the components of $\mathcal{G}$ partition $\mathcal{V}(\mathcal{G})$; Hence $\mathcal{V}\left(\mathcal{A}_{i}\right)$ 's, $i \in\left[k^{\prime}\right]$, partition $\mathcal{V}(\mathcal{G})$;
2. Throughout the Splitting Phase, any two different components of $\mathcal{G}$ are connected to each other by at most one edge in $\mathcal{G}$; Therefore, $\mathrm{s}_{\mathcal{G}}\left(\mathcal{V}\left(\mathcal{A}_{i}\right), \mathcal{V}\left(\mathcal{A}_{j}\right)\right) \in$ $\{0,1\}$ for every $i \neq j ;$
3. At any time during the Splitting Phase, the graph that is obtained from $\mathcal{G}$ by contracting the vertex set of each component of $\mathcal{G}$ to a single vertex is a tree; Therefore, $T^{\prime}$ is a tree;
4. In the Splitting Phase, every component of $\mathcal{G}$ is connected; Hence, as there are no bridges in each of the subgraph $\mathcal{A}_{i}$ for $i \in\left[k^{\prime}\right]$, they are all 2-edge connected.

It is also clear that each $\mathcal{A}_{i}$ belongs to a family in $\mathscr{P}$. Since $\mathcal{G}$ passes the Verifying Phase successfully, $\mathscr{T}^{\prime}=\left(\Gamma^{\prime}=\left[\mathcal{V}\left(\mathcal{A}_{1}\right), \ldots, \mathcal{V}\left(\mathcal{A}_{k^{\prime}}\right)\right], T^{\prime}\right)$ is already qualified to be a tree structure of Type $\operatorname{I}$ of $\mathcal{G}$. It remains to show that $\operatorname{mdc}\left(\mathscr{T}^{\prime}\right) \leq c$. Indeed, firstly, the root node $\mathcal{A}_{i_{0}}$ has at most $c$ connectors, hence at most $c$ downward connectors. Secondly, the other nodes $\mathcal{A}_{i}$ for $i \neq i_{0}$, each has a parent node and has at most $c+1$ connectors. Hence, each of them has at most $c \mathrm{DCs}$. Hence $\operatorname{mdc}\left(\mathscr{T}^{\prime}\right) \leq c$, as desired.

Claim 5.3.8. If $\mathcal{G} \in \mathscr{F}_{2}(c)$ then ALGO-2 terminates successfully.

Proof. Suppose $\mathcal{G} \in \mathscr{F}_{2}(c)$ and $\mathscr{T}=(\Gamma, T)$, where $\Gamma=\left[V_{1}, V_{2}, \ldots, V_{k}\right]$, is a relevant tree structure of $\mathcal{G}$. As each $\mathcal{G}_{i}$ is 2-edge connected, throughout the Splitting Phase,
the splits always occur at and only at the edges of $\mathcal{G}$ each of which connects two different nodes of $\mathscr{T}$. The Splitting Phase stops when and only when there are no more components that contain such an edge within each of them. Therefore, at the end of the Splitting Phase, the set of components of $\mathcal{G}$ generated by algorithm, namely $\left\{\mathcal{A}_{1}, \mathcal{A}_{2}, \ldots, \mathcal{A}_{k^{\prime}}\right\}$, coincide with the set of nodes $\left\{\mathcal{G}_{1}, \mathcal{G}_{2}, \ldots, \mathcal{G}_{k}\right\}$ of $\mathscr{T}$. The Verifying Phase also terminates successfully. Indeed, firstly, since each node $\mathcal{G}_{j}$ $(j \in[k])$ has at most $c$ downward connectors in $\mathscr{T}$, the node $\mathcal{A}_{i}(i \in[k])$ has at most $c+1$ connectors in $\mathscr{T}^{\prime}$. Secondly, a node $\mathcal{A}_{i_{0}}$ that has at most $c$ connectors always exists since we can choose, for instance, the root node of $\mathscr{T}$.

Claim 5.3.9. The running time of $A L G O-2$ is polynomial with respect to the order of $\mathcal{G}$.

Proof. Notice that every single task in ALGO-2 can be accomplished in polynomial time. Those tasks include: finding a min-cut of size one in a graph, deciding whether a graph belongs to $\mathscr{P}$, counting the number of connectors of each component of $\mathcal{G}$, and building a rooted tree based on the components of $\mathcal{G}$.

Let consider the "while loop" and the "for loop". After each intermediate iteration in the while loop, as at least one component gets split into two smaller components, the number of components of $\mathcal{G}$ is increased by at least one. Since the vertex sets of the components are pairwise disjoint, there are no more than $n=|\mathcal{V}(\mathcal{G})|$ components at any time. Hence, there are no more than $n$ iterations in the while loop. Since the number of graphs in $\mathcal{Q}_{1}$ cannot exceed $n$, the number of iterations in the for loop is also at most $n$. Thus, in total, the running time of ALGO-2 is polynomial with respect to $n$.

### 5.4 MinRanks of (Di)Graphs of Small Orders

To aid further research on the behavior of minranks of graphs, we have carried out a computation of binary minranks of all non-isomorphic graphs of orders up to 10 .

| Order | Number of non-isomorphic graphs | Total running time |
| :--- | :--- | :--- |
| 1 | 1 | $<1$ seconds |
| 2 | 2 | $<1$ seconds |
| 3 | 4 | $<1$ seconds |
| 4 | 11 | $<1$ seconds |
| 5 | 34 | $<1$ seconds |
| 6 | 156 | $<1$ seconds |
| 7 | 1,044 | $<1$ seconds |
| 8 | 12,346 | 25 seconds |
| 9 | 274,668 | 56 minutes |
| 10 | $12,005,168$ | 4.3 days |

Fig. 5.13: Running time for finding minranks of graphs or small orders

A reduction to the Satisfiability (SAT) problem [16] provides us with an elegant method to compute the binary minrank of a (hyper,di)graph. We observed that while the SAT-based approach is very efficient for (di)graphs having many edges, it does not perform well for simple instances, such as a graph on 10 vertices with no edges (minrank 10), or a digraph on seven vertices $0,1, \ldots, 6$, with six arcs $(0,1),(1,2),(2,3),(3,4),(4,5),(5,6)$ (minrank seven). For such naive instances, the SAT solver that we used, Minisat [24], was not able to terminate after hours. This weirdness might be attributable to the fact that the SAT instances corresponding to a (di)graph with fewer (arcs) edges contain more variables than those corresponding to a (di)graphs with more (arcs) edges on the same of set of vertices (see the Appendix A. 2 for more details).

To achieve our goal, we wrote a sub-program which used a Branch-and-Bound algorithm to find minranks in an exhaustive manner. When the input graph was
of large size, that is, its size surpassed a given threshold, a sub-program using a SAT solver was invoked; Otherwise, the Branch-and-Bound sub-program was used. We noticed that there were graphs of order 10 that have around 21-22 edges, for which the Branch-and-Bound sub-program could find the minranks in less than one second, while the SAT-based sub-program could not after 3 or 4 hours. For graphs of order 10 , we observed that the threshold 24 , which we actually used, did work well. The most time-consuming task is to compute the minranks of all $12,005,618$ non-isomorphic graphs of order 10. This task actually took more than four days to finish.

We put all details of the SAT-based approach [16] and the Branch-and-Bound approach in Appendix A. The minranks of all non-isomorphic graphs of orders at most six are listed in Appendix B for illustration. The minranks and the corresponding matrices that achieve the minranks of all non-isomorphic graphs of orders up to 10 are available at web.spms.ntu.edu.sg/~daus0001/mr-small-graphs.html. Computation of the minrank of a single (di)graph is available at web.spms.ntu. edu.sg/~daus0001/mr.html.

## 6. CONCLUSION

We have studied the security aspects and the computational aspects of the Index Coding with Side Information (ICSI) problem.

For block security, given a linear index code based on a matrix $\boldsymbol{L}$, we have established two bounds on the security level of the code. These bounds employ the minimum distance and the dual distance of $\mathcal{C}(\boldsymbol{L})$, the subspace spanned by the (transposed) columns of $\boldsymbol{L}$. While the dimension of this subspace, which is equal to the number of columns of $\boldsymbol{L}$ in this setting, corresponds to the number of transmissions in the scheme (that is, the efficiency of the code), the minimum distance determines the security of the scheme.

Open Problem I: Find a trade-off between the efficiency and the block security of a linear index code.

We now elaborate more on this problem. Consider an ICSI instance ( $m, n, \mathcal{X}, f$ ). Let $\mathcal{W}_{i} \triangleq\left\{c\left(\boldsymbol{u}^{(i)}+\boldsymbol{e}_{f(i)}\right): c \in \mathbb{F}_{q}^{*}, \boldsymbol{u}^{(i)} \triangleleft \mathcal{X}_{i}\right\}$ for $i \in[m]$. We have established (in Corollary 2.2.6) that $\boldsymbol{L}$ corresponds to a linear ( $m, n, \mathcal{X}, f$ )-IC over $\mathbb{F}_{q}$ if and only if $\mathcal{C}(\boldsymbol{L}) \cap \mathcal{W}_{i} \neq \varnothing$ for all $i \in[m]$. Here $\mathcal{C}(\boldsymbol{L})$ denotes the (transposed) column space of $\boldsymbol{L}$. Then the Open Problem I can be restated as follows: given an instance ( $m, n, \mathcal{X}, f$ ), what is the trade-off between the dimension and the minimum distance of a vector space $\mathcal{C} \subseteq \mathbb{F}_{q}^{n}$ that satisfies $\mathcal{C} \cap \mathcal{W}_{i} \neq \varnothing$ for all $i \in[m]$ ?

Open Problem II: Find constructions of families of linear index codes that possess good efficiency and good block security.

At one extreme, when only the efficiency is considered, it is known that designing an optimal linear index code is an NP-hard problem. At the other extreme, if we ignore the efficiency, we may ask the following question.

Open Problem III: How hard is it to find a linear index code that provides the highest level of block security?

We discuss below a different view of the Open Problem III. Let us consider the original setting of the ICSI problem [3], in which $q=2, m=n$, and $f(i)=i$ for all $i \in[n]$. Let $\boldsymbol{A}=\left(a_{i, j}\right)_{n \times n}$ be the binary side information matrix, defined by $a_{i, j}=1$ if and only if either $j=i \in[n]$ or $j \in \mathcal{X}_{i}$. Equivalently, $\boldsymbol{A}$ is obtained by taking the sum of the adjacency matrix of the side information digraph and the identity matrix. It is not hard to see that the task specified in the Open Problem III is equivalent to the task of finding a way to turn certain off-diagonal 1's into 0's in $\boldsymbol{A}$, so that the rows of the resulting matrix generate an error-correcting code of minimum distance as large as possible. Note that the row space of such a matrix corresponds to the transposed column space of a matrix $\boldsymbol{L}$ associated with a linear index code. It is very likely that this task is a hard problem. For comparison, even finding the minimum distance of an error-correcting code given its generating matrix corresponds to an NP-complete decision problem [65].

For strong security, we have introduced a lower bound on the length of a linear index code that is resistant to eavesdropping, information leaking, and errors. Index codes that achieve this bound have been constructed for sufficiently large alphabets. Open Problem IV: Investigate the strong security of linear index codes over small alphabets, such as the binary alphabet.

It seems that one cannot obtain the perfect strong security for linear index codes over small alphabets. However, we may consider the asymptotic strong security instead. In that case, vector linear index codes with block lengths approaching
infinity are more suitable. A valid question is: does the concatenation of binary wiretap codes (studied by Ozarow and Wyner [52]) and optimal binary vector linear index codes yield (asymptotically) optimal security levels and optimal transmission rates? Note that this question has been answered in the affirmative for the case of wiretap codes and scalar linear index codes that are over sufficiently large alphabets (Chapter 2 and 3).

We have also constructed a number of bounds on the length of an optimal errorcorrecting index code. As it is shown in Example 3.2.10, a separation of errorcorrecting code and index code sometimes leads to a non-optimal scheme. This raises a question of designing coding schemes in which the two layers are treated as a whole. We have discussed a general decoding procedure for linear error-correcting index codes. The difference between decoding of a classical error-correcting code and decoding of an error-correcting index code is that in the latter case, each receiver does not require a complete knowledge of the error vector. This difference may help to ease the decoding process. We state our sixth open problem below.

Open Problem VI: Find constructions of error-correcting index codes with good parameters and efficient decoding methods.

It has been shown that the minrank of the side information hypergraph is actually the length of an optimal scalar linear index code. However, it is, in general, extremely hard to compute the minrank of a hypergraph. Therefore, it would be desirable to find families of graphs, digraphs, and hypergraphs whose minranks can be determined in polynomial time. The second main contribution of this thesis is the revelation of some new families of (di)graphs for which the minrank computation can be done in polynomial time. We state below several open problems related to minranks of hypergraphs ((di)graphs).

Open Problem VII: Examine the hardness of the following decision problem:
decide whether a digraph has minrank two over a nonbinary field $\mathbb{F}_{q}$.
Open Problem VIII: Find a polynomial time algorithm to recognize a member of $\mathscr{F}_{1}(c)$ (see Section 5.3).

Open Problem IX: Find efficient exact algorithms for computing minrank of a hypergraph over a nonbinary field $\mathbb{F}_{q}$.

Open Problem X: Find new families of hypergraphs ((di)graphs) whose minranks can be found in polynomial time.

## BIBLIOGRAPHY

[1] R. Ahlswede, N. Cai, S. Y. R. Li, and R. W. Yeung. Network information flow. IEEE Trans. Inform. Theory, 46:1204-1216, 2000.
[2] N. Alon, A. Hassidim, E. Lubetzky, U. Stav, and A. Weinstein. Broadcasting with side information. In Proc. 49 th Annu. IEEE Symp. on Found. of Comput. Sci. (FOCS), pages 823-832, 2008.
[3] Z. Bar-Yossef, Z. Birk, T. S. Jayram, and T. Kol. Index coding with side information. In Proc. 47 th Annu. IEEE Symp. on Found. of Comput. Sci. (FOCS), pages 197-206, 2006.
[4] Z. Bar-Yossef, Z. Birk, T. S. Jayram, and T. Kol. Index coding with side information. IEEE Trans. Inform. Theory, 57(3):1479-1494, 2011.
[5] R. Bellman. The theory of dynamic programming. Bulletin of the American Mathematical Society, 60:503-516, 1954.
[6] C. H. Bennet, G. Brassard, and J. M. Robert. Privacy amplification by public discussion. SIAM J. Computing, 17:210-229, 1988.
[7] E. R. Berlekamp, R. J. McEliece, and H. C. A. van Tilborg. On the inherent intractability of certain coding problems. IEEE Trans. Inform. Theory, IT-24(3):384-386, 1978.
[8] Y. Berliner and M. Langberg. Index coding with outerplanar side information. In Proc. IEEE Symp. on Inform. Theory (ISIT), pages 869-873, Saint Petersburg, Russia, 2011.
[9] K. Bhattad and K. R. Narayanan. Weakly secure network coding. In Proc. 1st Workshop on Network Coding, Theory, and Applicat. (NetCod), 2005.
[10] Y. Birk and T. Kol. Informed-source coding-on-demand (ISCOD) over broadcast channels. In Proc. IEEE Conf. on Comput. Commun. (INFOCOM), pages 1257-1264, San Francisco, CA, 1998.
[11] Y. Birk and T. Kol. Coding-on-demand by an informed source (ISCOD) for efficient broadcast of different supplemental data to caching clients. IEEE Trans. Inform. Theory, 52(6):2825-2830, 2006.
[12] N. Cai and R. W. Yeung. Secure network coding. In Proc. IEEE Int. Symp. Inform. Theory (ISIT), Lausanne, Switzerland, 2002.
[13] C. Carlet. Vectorial Boolean Functions for Cryptography. In the book Boolean Models and Methods in Mathematics, Computer Science and Engineering, edited by Y. Crame and P. L. Hammer. Cambridge University Press, 2010.
[14] M. A. R. Chaudhry, Z. Asad, A. Sprintson, and M. Langberg. Finding sparse solutions for the index coding problem. In Proc. IEEE Globecom, pages 1-5, Houston, Texas, USA, 2011.
[15] M. A. R. Chaudhry, Z. Asad, A. Sprintson, and M. Langberg. On the complementary index coding problem. In Proc. IEEE Symp. on Inform. Theory (ISIT), pages 306-310, Saint Petersburg, Russia, 2011.
[16] M. A. R. Chaudhry and A. Sprintson. Efficient algorithms for index coding. In Proc. IEEE Conf. on Comput. Commun. (INFOCOM), pages 1-4, 2008.
[17] B. Chor, O. Goldreich, J. Hasted, J. Freidmann, S. Rudich, and R. Smolensky. The bit extraction problem or t-resilient functions. In Proc. 26th Annu. IEEE Symp. on Found. of Comput. Sci. (FOCS), pages 396-407, 1985.
[18] M. Chudnovsky, N. Robertson, P. Seymour, and R. Thomas. The strong perfect graph theorem. Annals of Mathematics, 164:51-229, 2006.
[19] G. Cornuejols, X. Liu, and K. Vuskovic. A polynomial time algorithm for recognizing perfect graphs. Proc. 48 th Annu. IEEE Symp. on Found. of Comput. Sci. (FOCS), pages 20-27, 2003.
[20] T. M. Cover and J. A. Thomas. Elements of information theory. WileyInterscience, New York, NY, USA, 1991.
[21] S. H. Dau, V. Skachek, and Y. M. Chee. Index coding and error correction. In Proc. IEEE Symp. on Inform. Theory (ISIT), pages 1753-1757, Saint Petersburg, Russia, 2011.
[22] S. H. Dau, V. Skachek, and Y. M. Chee. On secure index coding with side information. In Proc. IEEE Symp. on Inform. Theory (ISIT), pages 1046-1050, Saint Petersburg, Russia, 2011.
[23] C. Ding, R. Laihonen, and A. Renvall. Linear multisecret-sharing schemes and error-correcting codes. J. UCS, 3(9):1023-1036, 1997.
[24] N. Eén and N. Sörensson. An extensible SAT-solver. In Proc. 6th Int. Conf. Theory Appl. Satisf. Testing, 2003.
[25] S. El Rouayheb, M. A. R. Chaudhry, and A. Sprintson. On the minimum number of transmissions in single-hop wireless coding networks. In Proc. IEEE Inform. Theory Workshop (ITW), pages 120-125, 2007.
[26] S. El Rouayheb and E. Soljanin. On wiretap networks II. In Proc. IEEE Symp. on Inform. Theory (ISIT), pages 551-555, Nice, France, 2007.
[27] S. El Rouayheb, A. Sprintson, and C. Georghiades. On the relation between the index coding and the network coding problems. In Proc. IEEE Symp. on Inform. Theory (ISIT), pages 1823-1827, Toronto, Canada, 2008.
[28] S. El Rouayheb, A. Sprintson, and C. Georghiades. On the index coding problem and its relation to network coding and matroid theory. IEEE Trans. Inform. Theory, 56(7):3187-3195, 2010.
[29] J. Feldman, T. Malkin, C. Stein, and R. A. Servedio. On the capacity of secure network coding. In Proc. Annu. Allerton Conf. Commun., Control, and Comput., 2004.
[30] A. Frank and A. Gyarfas. Directed graphs and computer programs. Problemes Combinatoires et Theorie des Graphes, (260):157-158, 1976.
[31] M. Grassl. Bounds on the minimum distance of linear codes and quantum codes. available online at http://www.codetables.de.
[32] B. Guenin and R. Thomas. Packing directed circuits exactly. Manuscript, 2001.
[33] K.C. Gupta and P. Sarkar. Improved construction of nonlinear resilient s-boxes. IEEE Trans. Inform. Theory, 51(1):339-348, 2005.
[34] W. Haemers. An upper bound for the Shannon capacity of a graph. Algebr. Methods Graph Theory, 25:267-272, 1978.
[35] F. Harary and R. Norman. Some properties of line digraphs. Rendiconti del Circolo Matematico Palermo, 9:149-163, 1960.
[36] I. Haviv and M. Langberg. On linear index coding for random graphs. http://arxiv.org/abs/1107.0390, 2011.
[37] A. S. Hedayat, N. J. A. Sloane, and J. Stufken. Orthogonal Arrays: Theory and Applications. Springer-Verlag, New York, 1999.
[38] J. Hopcroft and R. Tarjan. Efficient planarity testing. J. ACM, 21(4):549-568, 1974.
[39] W. Huffman and V. Pless. Fundamentals of Error-Correcting Codes. Cambridge University Press, 2003.
[40] R. M. Karp. Reducibility among combinatorial problems. Complexity Comput. Comput., 40(4):85-103, 1972.
[41] S. Katti, D. Katabi, H. Balakrishnan, and M. Médard. Symbol-level network coding for wireless mesh networks. ACM SIGCOMM Comput. Commun. Review, 38(4):401-412, 2008.
[42] S. Katti, H. Rahul, W. Hu, D. Katabi, M. Médard, and J. Crowcroft. Xors in the air: Practical wireless network coding. In Proc. ACM SIGCOMM, pages 243-254, 2006.
[43] R. Koetter and M. Médard. An algebraic approach to network coding. IEEE/ACM Trans. Netw., 11:782-795, 2003.
[44] M. Langberg and A. Sprintson. On the hardness of approximating the network coding capacity. In Proc. IEEE Symp. on Inform. Theory (ISIT), pages 315319, Toronto, Canada, 2008.
[45] L. Lovász. On the Shannon capacity of a graph. IEEE Trans. Inform. Theory, 25:1-7, 1979.
[46] E. Lubetzky and U. Stav. Non-linear index coding outperforming the linear optimum. Proc. 48 th Annu. IEEE Symp. on Found. of Comput. Sci. (FOCS), pages 161-168, 2007.
[47] C. Lucchesi. A minimax equality for directed graphs. PhD thesis, University of Waterloo, Ontario, 1976.
[48] C. Lucchesi and D. Younger. A minimax theorem for directed graphs. J. London Math. Soc., 17(2):369-374, 1978.
[49] F. J. MacWilliams and N. J. A. Sloane. The Theory of Error-Correcting Codes. Amsterdam: North-Holland, 1977.
[50] J. L. Massey. Minimal codewords and secret sharing. In Proc. Joint SwedishRussian Int. Workshop on Inform. Theory, pages 276-279, 1993.
[51] B. D. McKay. Practical graph isomorphism. Congressus Numerantium, 30:4587, 1981.
[52] L. H. Ozarow and A. D. Wyner. The wire-tap channel II. Bell Syst. Tech. J., 63:2135-2157, 1984.
[53] R. Peeters. Orthogonal representations over finite fields and the chromatic number of graphs. Combinatorica, 16(3):417-431, 1996.
[54] V. Ramachandran. A minimax arc theorem for reducible flow graphs. SIAM J. Discrete Math., pages 554-560, 1990.
[55] P. D. Seymour. Packing circuits in Eulerian digraphs. Combinatorica, 16(2):223-231, 1996.
[56] A. Shamir. A linear time algorithm for finding minimum cutsets in reducible graphs. J. Comput., 8(4):645-655, 1979.
[57] C. E. Shannon. The zero-error capacity of a noisy channel. IRE Trans. Inform. Theory, 3:3-15, 1956.
[58] D. Silva and F. R. Kschischang. Universal weakly secure network coding. In Proc. Inform. Theory Workshop on Networking and Inform. Theory (ITW), pages 281-285, 2009.
[59] D. Silva and F. R. Kschischang. Universal secure error-correcting schemes for network coding. In Proc. IEEE Symp. on Inform. Theory (ISIT), pages 24282432, Austin, Texas, USA, 2010.
[60] M. Syslo. A labeling algorithm to recognize a line digraph and output its root graph. Inform. Process. Lett., 15(1):28-30, 1982.
[61] J. L. Szwarcfiter. On minimum cuts of cycles and maximum disjoint cycles. In Graphs and algorithms, vol. 89 of Contemp. Math., 153-166. Amer. Math. Soc., Providence, RI, 1989.
[62] R. E. Tarjan. Depth-first search and linear graph algorithms. SIAM J. Comput., $1(2): 146-160,1972$.
[63] R. E. Tarjan. Testing flow graph reducibility. J. Comput. System Sci., (9):355365, 1974.
[64] G. S. Tseitin. On the complexity of derivation in propositional calculus. In the book Studies in Constructive Mathematics and Mathematical Logic, Part 2, edited by A. O. Slisenko. Consultants Bureau, New York, 1970.
[65] A. Vardy. The intractability of computing the minimum distance of a code. IEEE Trans. Inform. Theory, 43(6):1757-1766, 1997.
[66] C.-C. Wang, E. L. Lloyd, and M. L. Soffa. Feedback vertex sets and cyclically reducible graphs. J. ACM, 32(2):296-313, 1985.
[67] D. B. West. Introduction to graph theory. Prentice Hall, 2 edition, 1996.
[68] M. Wiegers. Recognizing outerplanar graphs in linear time. In Int. Workshop WG '86 on Graph-Theoretic Concepts in Comput. Sci., pages 165-176, 1987.
[69] Y. Wu, J. Padhye, R. Chandra, V. Padmanabhan, and P. A. Chou. The local mixing problem. In Proc. Inform. Theory and Applicat. Workshop, San Diego, CA, 2006.
[70] X-M. Zhang and Y. Zheng. On nonlinear resilient functions. In Proc. 14 th Annu. Int. Conf. on Theory and Appl. of Cryptographic Tech. (EUROCRYPT), pages 274-288, 1995.
[71] Z. Zhuang, Y. Luo, and A. J. Han Vinck. Secure error-correcting network codes with side information from source. In Proc. Int. Conf. Commun. Intell. Inform. Secur., pages 55-59, 2010.

## APPENDIX

# A. AN EXACT ALGORITHM FOR FINDING MINRANKS OF (DI)GRAPHS OVER $\mathbb{F}_{2}$ 

## A. 1 The Algorithm

We described below the algorithm for digraphs. For graphs, it is completely the same.

```
Main()
{
Input a digraph \mathcal{D and a threshold }0\mathrm{ ;}
if }|\mathcal{E}(\mathcal{D})|\leq0\mathrm{ then
    Find-MinRank-Using-BranchAndBound(\mathcal{D});
else
    Find-MinRank-Using-SAT-Solver(\mathcal{D});
end if
}
```

Fig. A.1: Main module

The Branch-and-Bound ( BB ) module runs through the set of all matrices that fit $\mathcal{D}$ (possibly ignores some) and finds out a matrix that has smallest minrank. Suppose $\mathcal{D}$ is of order $n$.

The BB module builds up a search tree as follows. The nodes of the tree are grouped into $n+1$ different layers. The root node belongs to Layer 0 . The children of the root node belong to Layer 1 . The children of each of the children of the root node belong to Layer 2, and so forth. Each node in Layer $i(i \in[n])$ is labeled by a row vector $\boldsymbol{u}=\left(u_{1}, u_{2}, \ldots, u_{n}\right) \in \mathbb{F}_{2}^{n}$ satisfying $u_{i}=1$ and $u_{j}=0$ if $(i, j) \notin \mathcal{E}(\mathcal{D})$.

In other words, the label of each node in Layer $i$ is a candidate for the $i$ th row of a matrix that fits $\mathcal{D}$. The sets of labels of the children of each node in Layer $i-1$ $(2 \leq i \leq n)$ is precisely the set of all such row vectors $\boldsymbol{u}$. The $n$ labels of the nodes on an arbitrary path from the root to a leaf (excluding the root) form the rows of a matrix that fits $\mathcal{D}$. Traversing all such paths of the search tree, we can find all matrices that fit $\mathcal{D}$. The following example illustrates the search tree described above.


Fig. A.2: An example of the search tree built by the BB module

The BB module during its running time maintains a global bound, which is a "current best" candidate for the minrank of $\mathcal{D}$. This bound is the rank of a matrix that fits $\mathcal{D}$ and has lowest rank among all previously examined matrices. The global bound is updated every time a matrix with lower rank is examined. Based on this bound, the algorithm may prune the branches (i.e., it bypasses those branches and goes upwards to examine other branches) that surely do not contain matrices with lower ranks than the current best rank. Hence, the search space can be reduced significantly. At the beginning, we can initiate this bound to be any upper bound
for $\operatorname{minrk}_{2}(\mathcal{D})$, found by some fast heuristic algorithm. Some candidates are those established by Chaudhry and Sprintson [16]. In the case of graphs, we actually used a simple upper bound for $\operatorname{cc}(\mathcal{G})$ (hence it is also an upper bound for minrk $_{2}(\mathcal{G})$ ), which was obtained by coloring $\overline{\mathcal{G}}$ greedily. Note that we should choose different heuristic algorithms in different scenarios. Indeed, if we want to compute the min-rank of only one (di)graph, an algorithm that outputs a good bound in half a minute seems to be fine. However, if we are computing the minranks of millions of (di)graphs, such an algorithm would be a very poor choice. In such a case, an algorithm that quickly produces a not-so-good bound might be preferred.

```
Find-MinRank-Using-SAT-Solver ( \(\mathcal{D}\) )
\{
lower-bound \(:=1\); upper-bound \(:=|\mathcal{V}(\mathcal{D})|\);
while upper-bound \(>\) lower-bound +1 do
    \(k:=\lfloor(\) upper-bound + lower-bound \() / 2\rfloor ;\)
    Create the corresponding SAT instance \(I_{k}(\mathcal{D})\);
    sat \(:=\operatorname{SAT}-\operatorname{Solver}\left(I_{k}(\mathcal{D})\right)\);
    if sat \(=\) SATISFIABLE then
        upper-bound \(:=k\);
    else
        lower-bound :=k;
    end if
end while
Print out upper-bound as the minrank of \(\mathcal{D}\);
\}
```

Fig. A.3: SAT-based module

The issue of creating the SAT instance $I_{k}(\mathcal{D})$ is discussed in details in the next section.

## A. 2 Reduction to SAT Problem

In this section, we discuss a reduction of the ICSI problem to the SAT problem (Fig. A.3). The reduction is attributed to Chaudhry and Sprintson [16]. Although the reduction works for ICSI instances described by hypergraphs, we here restrict ourselves to the case of (di)graphs, and provide more details on the implementation of the reduction.

Suppose the digraph $\mathcal{D}$ describes the following ICSI instance. There are $n$ receivers $R_{1}, R_{2}, \ldots, R_{n}$, and $n$ messages $x_{1}, x_{2}, \ldots, x_{n}$ in $\mathbb{F}_{2}$. The receiver $R_{i}(i \in[n])$ possesses $\left\{x_{j}\right\}_{j \in \mathcal{X}_{i}}$ as side information, and requests $x_{i}$. Given a value $k>0$, based on a SAT solver, we want to decide whether the sender can satisfy the demands from all receivers by broadcasting at most $k$ linear combinations of the $x_{i}$ 's.

Suppose the $k$ encoding vectors are

$$
\boldsymbol{g}_{j}=\left(g_{1, j}, g_{2, j}, \ldots, g_{n, j}\right) \in \mathbb{F}_{2}^{n}, j \in[k] .
$$

The sender broadcasts $k$ bits

$$
\boldsymbol{x} \boldsymbol{g}_{j}^{T}=\sum_{i=1}^{n} x_{i} g_{i, j}, \quad j \in[k] .
$$

Suppose $R_{i}(i \in[n])$ uses the decoding vector

$$
\boldsymbol{q}_{i}=\left(q_{i, 1}, q_{i, 2}, \ldots, q_{i, k}\right) \in \mathbb{F}_{2}^{k}
$$

He applies the decoding vector to the received bits

$$
\boldsymbol{q}_{i}\left(\boldsymbol{x} \boldsymbol{g}_{1}^{T}, \boldsymbol{x} \boldsymbol{g}_{2}^{T}, \ldots, \boldsymbol{x} \boldsymbol{g}_{k}^{T}\right)^{T}=\sum_{r=1}^{n}\left(\sum_{j=1}^{k} g_{r, j} q_{i, j}\right) x_{r} .
$$

The receiver $R_{i}$ can retrieve $x_{i}$ successfully if and only if

$$
\left\{\begin{array}{l}
\sum_{j=1}^{k} g_{i, j} q_{i, j}=1, \\
\sum_{j=1}^{k} g_{r, j} q_{i, j}=0, \quad \forall r \notin\{i\} \cup \mathcal{X}_{i} .
\end{array}\right.
$$

If we regard 0 as FALSE and 1 as TRUE, then the summation and the product in $\mathbb{F}_{2}$ are equivalent to the $\mathrm{XOR} \oplus$ and the $\mathrm{AND} \wedge$ operators in Boolean logic, respectively. Therefore, it is possible for the sender to satisfy all demands with $k$ linear combinations of the messages if and only if the following Boolean formula is satisfiable

$$
\begin{equation*}
B_{k} \triangleq \bigwedge_{i=1}^{n}\left\{\left[\bigoplus_{j=1}^{k}\left(g_{i, j} \wedge q_{i, j}\right)\right] \bigwedge\left\{\bigwedge_{r \notin\{i\} \cup \mathcal{X}_{i}} \overline{\left[\bigoplus_{j=1}^{k}\left(g_{r, j} \wedge q_{i, j}\right)\right]}\right\}\right\} \tag{A.1}
\end{equation*}
$$

Most of the current SAT solvers require the input to be in CNF (Conjunctive Normal Form), that is, the input formula should be a conjunction of clauses

$$
C_{1} \wedge C_{2} \wedge \cdots \wedge C_{s}
$$

where each clause $\mathcal{C}_{i}$ is a disjunction of literals

$$
C_{i}=L_{i, 1} \vee L_{i, 2} \vee \cdots \vee L_{i, r},
$$

and each literal is either a plain variable or a negated variable.
Any Boolean formula can be transformed into an equivalent formula, that is, a formula with the same set of variables and the same satisfiability property, based on rules about logical equivalences. However, a straightforward transformation can lead to an exponential explosion of the formula, that is, the number of clauses is
increased exponentially. To avoid the exponential blowup in the size (number of variables plus number of clauses) of the formula, the Tseitin transformation is often used. The Tseitin transformation [64] transforms an arbitrary Boolean formula into a new one such that not only the satisfiability is preserved but also the size of the new formula is linear with respect to the size of the original one. Given a formula $\phi$, the idea is to introduce a fresh variable $\tau_{\theta}$ for every sub-formula $\theta$ of $\phi$. Then one adds constraints in CNF which ensure that $\tau_{\theta} \longleftrightarrow \theta$ holds. Finally, one adds $\wedge \tau_{\phi}$ to the formula to make sure that the satisfiability is preserved.

The Tseitin transformations of the basic formulas that we need are

$$
\begin{align*}
& \tau \longleftrightarrow(a \wedge b) \rightsquigarrow(a \vee \bar{\tau}) \wedge(b \vee \bar{\tau}) \wedge(\bar{a} \vee \bar{b} \vee \tau), \\
& \tau \longleftrightarrow(\bar{a}) \rightsquigarrow(a \vee \tau) \wedge(\bar{a} \vee \bar{\tau}),  \tag{A.2}\\
& \tau \longleftrightarrow(a \oplus b) \rightsquigarrow(a \vee b \vee \bar{\tau}) \wedge(\bar{a} \vee \bar{b} \vee \bar{\tau}) \wedge(a \vee \bar{b} \vee \tau) \wedge(\bar{a} \vee b \vee \tau)
\end{align*}
$$

For each $i \in[n]$ we consider the following formula


By introducing fresh variables $h_{i, j}$ 's and $s_{i, j}$ 's for the sub-formulas of the formula
above, we obtain a new formula

$$
\begin{aligned}
& \left(h_{i, 1} \longleftrightarrow\left(g_{i, 1} \wedge q_{i, 1}\right)\right) \wedge \cdots \wedge\left(h_{i, k} \longleftrightarrow\left(g_{i, k} \wedge q_{i, k}\right)\right) \\
\wedge & \left(s_{i, 1} \longleftrightarrow\left(h_{i, 1} \oplus h_{i, 2}\right)\right) \wedge\left(s_{i, 2} \longleftrightarrow\left(s_{i, 1} \oplus h_{i, 3}\right)\right) \wedge \cdots \wedge\left(s_{i, k-1} \longleftrightarrow\left(s_{i, k-2} \oplus h_{i, k}\right)\right) \\
\wedge & s_{i, k-1}
\end{aligned}
$$

We now apply (A.2) to the above formula and obtain a formula in CNF

$$
\begin{aligned}
C_{i} \triangleq & \left(g_{i, 1} \vee \overline{h_{i, 1}}\right) \wedge\left(q_{i, 1} \vee \overline{h_{i, 1}}\right) \wedge\left(\overline{g_{i, 1}} \vee \overline{q_{i, 1}} \vee h_{i, 1}\right) \\
& \wedge\left(g_{i, 2} \vee \overline{h_{i, 2}}\right) \wedge\left(q_{i, 2} \vee \overline{h_{i, 2}}\right) \wedge\left(\overline{g_{i, 2}} \vee \overline{q_{i, 2}} \vee h_{i, 2}\right) \\
& \vdots \\
& \wedge\left(g_{i, k} \vee \overline{h_{i, k}}\right) \wedge\left(q_{i, k} \vee \overline{h_{i, k}}\right) \wedge\left(\overline{g_{i, k}} \vee \overline{q_{i, k}} \vee h_{i, k}\right) \\
& \wedge\left(h_{i, 1} \vee h_{i, 2} \vee \overline{s_{i, 1}}\right) \wedge\left(\overline{h_{i, 1}} \vee \overline{h_{i, 2}} \vee \overline{s_{i, 1}}\right) \wedge\left(h_{i, 1} \vee \overline{h_{i, 2}} \vee s_{i, 1}\right) \wedge\left(\overline{h_{i, 1}} \vee h_{i, 2} \vee s_{i, 1}\right) \\
& \wedge\left(h_{i, 3} \vee s_{i, 1} \vee \overline{s_{i, 2}}\right) \wedge\left(\overline{h_{i, 3}} \vee \overline{s_{i, 1}} \vee \overline{s_{i, 2}}\right) \wedge\left(h_{i, 3} \vee \overline{s_{i, 1}} \vee s_{i, 2}\right) \wedge\left(\overline{h_{i, 3}} \vee s_{i, 1} \vee s_{i, 2}\right) \\
& \vdots \\
& \wedge\left(h_{i, k} \vee s_{i, k-2} \vee \overline{s_{i, k-1}}\right) \wedge\left(\overline{h_{i, k}} \vee \overline{s_{i, k-2}} \vee \overline{s_{i, k-1}}\right) \wedge\left(h_{i, k} \vee \overline{s_{i, k-2}} \vee s_{i, k-1}\right) \\
& \wedge\left(\overline{h_{i, k}} \vee s_{i, k-2} \vee s_{i, k-1}\right) \\
& \wedge s_{i, k-1} .
\end{aligned}
$$

Similarly, for each $i \in[n]$ and $r \notin\{i\} \cup \mathcal{X}_{i}$ we consider the following formula

By introducing fresh variables $t_{i, r, j}$ 's and $u_{i, r, j}$ 's to the sub-formulas of this formula, we obtain a new formula

$$
\begin{aligned}
D_{i, r} \triangleq & \left(t_{i, r, 1} \longleftrightarrow\left(g_{r, 1} \wedge q_{i, 1}\right)\right) \wedge \cdots \wedge\left(t_{i, r, k} \longleftrightarrow\left(g_{r, k} \wedge q_{i, k}\right)\right) \\
& \wedge\left(u_{i, r, 1} \longleftrightarrow\left(t_{i, r, 1} \oplus t_{i, r, 2}\right)\right) \wedge\left(u_{i, r, 2} \longleftrightarrow\left(u_{i, r, 1} \oplus t_{i, r, 3}\right)\right) \wedge \cdots \\
& \wedge\left(u_{i, r, k-1} \longleftrightarrow\left(u_{i, r, k-2} \oplus t_{i, r, k}\right)\right) \\
& \wedge\left(u_{i, r, k} \longleftrightarrow \overline{u_{i, r, k-1}}\right) \\
& \wedge u_{i, r, k}
\end{aligned}
$$

We can rewrite $D_{i, r}$ in CNF as follows

$$
\begin{aligned}
D_{i, r}= & \left(g_{r, 1} \vee \overline{t_{i, r, 1}}\right) \wedge\left(q_{i, 1} \vee \overline{t_{i, r, 1}}\right) \wedge\left(\overline{g_{r, 1}} \vee \overline{q_{i, 1}} \vee t_{i, r, 1}\right) \\
& \wedge\left(g_{r, 2} \vee \overline{t_{i, r, 2}}\right) \wedge\left(q_{i, 2} \vee \overline{t_{i, r, 2}}\right) \wedge\left(\overline{g_{r, 2}} \vee \overline{q_{i, 2}} \vee t_{i, r, 2}\right) \\
& \vdots \\
& \wedge\left(g_{r, k} \vee \overline{t_{i, r, k}}\right) \wedge\left(q_{i, k} \vee \overline{t_{i, r, k}}\right) \wedge\left(\overline{g_{r, k}} \vee \overline{q_{i, k}} \vee t_{i, r, k}\right) \\
& \wedge\left(t_{i, r, 1} \vee t_{i, r, 2} \vee \overline{u_{i, r, 1}}\right) \wedge\left(\overline{t_{i, r, 1}} \vee \overline{t_{i, r, 2}} \vee \overline{u_{i, r, 1}}\right) \wedge\left(t_{i, r, 1} \vee \overline{t_{i, r, 2}} \vee u_{i, r, 1}\right) \\
& \wedge\left(\overline{t_{i, r, 1}} \vee t_{i, r, 2} \vee u_{i, r, 1}\right) \\
& \wedge\left(t_{i, r, 3} \vee u_{i, r, 1} \vee \overline{u_{i, r, 2}}\right) \wedge\left(\overline{t_{i, r, 3}} \vee \overline{u_{i, r, 1}} \vee \overline{u_{i, r, 2}}\right) \wedge\left(t_{i, r, 3} \vee \overline{u_{i, r, 1}} \vee u_{i, r, 2}\right) \\
& \wedge\left(\overline{t_{i, r, 3}} \vee u_{i, r, 1} \vee u_{i, r, 2}\right) \\
& \therefore \\
& \wedge\left(t_{i, r, k} \vee u_{i, r, k-2} \vee \overline{u_{i, r, k-1}}\right) \wedge\left(\overline{t_{i, r, k}} \vee \overline{u_{i, r, k-2}} \vee \overline{u_{i, r, k-1}}\right) \wedge\left(t_{i, r, k} \vee \overline{u_{i, r, k-2}} \vee u_{i, r, k-1}\right) \\
& \wedge\left(u_{i, r, k-1} \vee u_{i, r, k}\right) \wedge\left(\overline{u_{i, r, k-1}} \vee \overline{u_{i, r, k}}\right) \\
& \wedge u_{i, r, k} .
\end{aligned}
$$

We obtain the following formula (in CNF)

$$
B_{k}^{\prime}=\bigwedge_{i=1}^{n}\left\{C_{i} \bigwedge\left(\bigwedge_{r \notin\{i\} \cup \mathcal{X}_{i}} D_{i, r}\right)\right\}
$$

which is a Tseitin transformation of the formula $B_{k}$ given in (A.1). Therefore, $B_{k}^{\prime}$ is satisfiable if and only if $B_{k}$ is satisfiable. Thus, the sender can satisfy all requests from the receivers by sending out at most $k$ linear combinations of the messages if and only if $B_{k}^{\prime}$ is satisfiable. Since $B_{k}^{\prime}$ is in CNF, its satisfiability can be tested using any SAT solver.

The formula $B^{\prime}$ has $n(2 k n+2 k-1)-2 k \sum_{i=1}^{n}\left|\mathcal{X}_{i}\right|$ variables and $n(7 k n-n-$ 2) $-(7 k-1) \sum_{i=1}^{n}\left|\mathcal{X}_{i}\right|$. Observe that $\sum_{i=1}^{n}\left|\mathcal{X}_{i}\right|$ is the number of arcs in the case of digraphs and is two times the number of edges in the case of graphs. Therefore, for two (di)graphs of the same order, the (di)graph with fewer (arcs) edges produces a CNF formula consisting of more variables. That might explain why the SAT-based algorithm seems to be not very good at dealing with (di)graphs having very few (arcs) edges.

## A. 3 Generating Lists of Non-Isomorphic Graphs

In order to compute the minranks of all graphs of certain order, we first need to produce a list of all non-isomorphic graphs of that order. The Nauty package, written by McKay [51], is a good tool for that purpose. Once the package (http: //cs.anu.edu.au/~bdm/nauty/) is installed, we can first use the command (in Linux)
./geng $n$ graphs-n-vertices.txt
to generate a list of all non-isomorphic graphs over $n$ vertices and write the list to the file graphs-n-vertices.txt. If we also use the option -l then the graphs are canonically labeled. The graphs in this list are written in the compact graph6 format. If we want to write them in a readable format, we can call

> ./listg -q -e graphs-n-vertices.txt graphs-n-vertices-readable.txt

Then in the file graphs-n-vertices-readable.txt, each graph will be written in the following format

$$
n m a_{1} b_{1} a_{2} b_{2} \cdots a_{m} b_{m}
$$

where $n$ and $m$ are the number of vertices and edges, respectively, and $\left\{a_{i}, b_{i}\right\}$ is the $i$ th edge of the graph. For more information on how to use other functionalities of the Nauty package, please refer to its user guide.

## B. MINRANKS OF GRAPHS OF SMALL ORDERS

B. 1 MinRanks of Graphs of Orders One to Four

Graphs of minrank one:


Graphs of minrank two:

- $\quad$



Graphs of minrank three:


Graphs of minrank four:


## B. 2 MinRanks of Graphs of Order Five

Graphs of minrank one:


Graphs of minrank two:







Graphs of minrank three:
$\rightarrow$





-



 4



Graphs of minrank four:


Graphs of minrank five:


## B. 3 MinRanks of Graphs of Order Six

Graphs of minrank one:


Graphs of minrank two:

$\otimes \Delta \Delta$
$\otimes \otimes \otimes$


*     * 

$\otimes *$
$* * *$
＊

$\vec{\Delta} \vec{\Delta}$
』 4＊＊
$甘 \forall 甘 甘$
$甘 * 44$
$11 * 4 \sqrt{*}$
N $4 \otimes$
$\otimes \otimes \mathbb{N}$
$\mathbb{N} \otimes \otimes$

\# * $\otimes *$

*     *         * 
* $\downarrow$ * $\downarrow$
$\Delta * * *$
 $\star * * *$ * * * $\otimes * * *$
$\otimes * *$
*     * 

$\otimes * * *$ 0
$\because \dot{A} \dot{A}$

$\mathbb{N} * *$
$N W \pm \otimes$
$\forall N W W$
$\pm W * W$
$\otimes \otimes \otimes \otimes$


Graphs of minrank five:


Graphs of minrank six:


## INDEX

| $(\rho, \delta)$-Property, 83 | $\mathcal{X}, 22$ |
| :--- | :--- |
| 2-edge connected, 17 | $\mathcal{X}_{A}, 26$ |
| $N_{O}^{\mathcal{D}}(u), 18$ | $\mathcal{X}_{i}, 22$ |
| $N^{\mathcal{G}}(u), 18$ | $\mathcal{Y}_{i}, 56$ |
| $N_{O}(u)$, see $N_{O}^{\mathcal{D}}(u)$ | $\mathcal{Z}, 42$ |
| $N_{q}(M, d), 14$ | $\mathcal{Z}_{i}, 42$ |
| $N_{q}[k, d], 14$ | $\alpha(\mathcal{D}), 62$ |
| $[n], 13$ | $\alpha(\mathcal{G}), 18$ |
| $\mathscr{F}_{1}(c), 129$ | $\alpha$-bound, 63,94 |
| $\mathscr{F}_{2}(c), 129$ | $\alpha(\mathcal{H}), 61$ |
| $\mathcal{F}(m, n, \mathcal{X}, \mathcal{Z}, f), 42$ | $\boldsymbol{e}_{i}, 14$ |
| $\mathcal{G}-E, 130$ | $\mathcal{C}(\boldsymbol{L}), 25$ |
| $\mathcal{G}-V, 130$ | $\mathrm{H}(\cdot \mid \cdot)$, see conditional entropy |
| $\mathcal{G}-e, 130$ | $\mathrm{H}(\cdot)$, see entropy |
| $\mathcal{G}-v, 130$ | $\mathfrak{J}(\Gamma), 82$ |
| $\Gamma(n, \rho), 83$ | $\mathbb{F}_{q}, 12$ |
| $\mathcal{I}(q, \mathcal{H}), 56$ | $\mathbb{F}_{q}^{*}, 12$ |
| $\mathcal{J}(\mathcal{H}), 56$ | $\kappa$-bound, 65 |
| $\mathcal{L}_{i}, 77$ | $\kappa_{q}, 34$ |
| $\mathcal{N}_{q}[\delta, \mathcal{H}], 56$ | $\triangleleft, 13$ |
| $\mathscr{P}, 127$ | mdc $(\mathscr{T}), 129$ |
| $\mathscr{T}, 128$ | minrk, 19 |

$\nu_{0}(\mathcal{D}), 115$
$\nu_{1}(\mathcal{D}), 118$
$\bar{\kappa}_{q}, 43$
$\mathcal{S}_{i}, 129$
$\mathrm{s}_{\mathcal{G}}(U, V), 127$
$\operatorname{span}_{q}, 15$
$\tau_{0}(\mathcal{D}), 118$
$\tau_{1}(\mathcal{D}), 118$
$\widehat{\mathcal{X}}_{A}, 28$
$n(\Gamma), 82$
$\mathcal{H}$-IC over $\mathbb{F}_{q}$, see index code
$(m, n, \mathcal{X}, f)$-IC over $\mathbb{F}_{q}$, see index code
acyclic, 17
adjacent, 15
adversary, 27
strength, 27
arc, 15
in-coming, 15
out-going, 15
bridge, 17
circuit, 17
circuit packing number, 115
circuit-packing bound, 115
clique, 18
clique cover, 18
minimum, 18
number, 18
clique-cover bound
digraph, 116
graph, 67
code
$(n, M, d)_{q}, 14$
$[n, k, d]_{q}, 13$
$[n, k]_{q}, 13$
dimension, 13
generator matrix, 13
length, 13
linear, 13
MDS, 14
parity-check matrix, 14
size, 14
codeword, 13
coloring, 100
fair, 100
conditional entropy, 20
confusable, 92
connected component, 17
connector
downward, 129
upward, 129
Construction
A, 50
endpoint, 15
B, 75
entropy, 19
cut, 17
cycle, 17

DC, see connector, downward
digraph, 15
complement, 16
complete, 16
connectively reducible, 119
fully reducible flow, 119
order, 15
partially planar, 122
planar, 122
size, 15
strongly connected, 17
symmetric, 16
that pack, 119
underlying graph, 16
distance
dual, 14
Hamming, 13

ECIC, 55
$(\delta, \mathcal{H})-, 55$
optimal, 56
edge, 15
hyperarc, 19
hypergraph, 22
underlying digraph, 23
underlying graph, 24
IC, see index code

ICSI instance, 22
ICSRI, 42
independence number, 18
independent set, 18
generalized, 61
maximum, 18
maximum generalized, 61
index code, 24
based on $\boldsymbol{L}, 25$
block length, 125
error-correcting, 55
length, 24
linear, 24
matrix corresponding to, 25
optimal, 24
randomized, 44
static, 82
transmission rate, 125
line digraph, 119
minrank
digraph, 19
graph, 19
hypergraph, 34
no information, 27
node, 128
child, 129
leaf, 129
parent, 129
root, 129
out-degree, 18
out-neighbor, 18
path, 17
Pr, 19
relevant error pattern, 77
relevant tree structure, 129
resilient function, 84
weakly, 85
security
block, 27
completely insecure, 28
strong, 44
weak, 27
side information
digraph, 23
graph, 24
hypergraph, 22
Singleton bound, 14, 69, 95
strongly connected component, 17
subgraph, 16
induced, 17
maximum acyclic induced, 62
support, 13
tree, 18
rooted, 18
tree structure of Type I, 128

UC, see connector, upward
vertex, 15
child, 18
parent, 18
weight, 13

