entropy

MDPI

Article

Index Coding with Multiple Interpretations

Valéria G. Pedrosa * and Max H. M. Costa *

School of Electrical and Computer Engineering, University of Campinas, Campinas 13083-852, SP, Brazil
* Correspondence: v230000@dac.unicamp.br (V.G.P.); max@fee.unicamp.br (M.H.M.C.)

Abstract: The index coding problem consists of a system with a server and multiple receivers with different side information and demand sets, connected by a noiseless broadcast channel. The server knows the side information available to the receivers. The objective is to design an encoding scheme that enables all receivers to decode their demanded messages with a minimum number of transmissions, referred to as an index code length. The problem of finding the minimum length index code that enables all receivers to correct a specific number of errors has also been studied. This work establishes a connection between index coding and error-correcting codes with multiple interpretations from the tree construction of nested cyclic codes. The notion of multiple interpretations using nested codes is as follows: different data packets are independently encoded, and then combined by addition and transmitted as a single codeword, minimizing the number of channel uses and offering error protection. The resulting packet can be decoded and interpreted in different ways, increasing the error correction capability, depending on the amount of side information available at each receiver. Motivating applications are network downlink transmissions, information retrieval from datacenters, cache management, and sensor networks.

Keywords: index coding; pliable index coding; error correcting index coding

Citation: Pedrosa, V.G.; Costa, M.H.M. Index Coding with Multiple Interpretations. *Entropy* **2022**, *24*, 1149. https://doi.org/10.3390/e24081149

Academic Editor: Sangun Park

Received: 20 June 2022
Accepted: 16 August 2022
Published: 18 August 2022

1. Introduction

In this work, we consider a source code variant, introduced by Birk and Kol [1], originally called informed source coding-on-demand (ISCOD), and further developed by Bar-Yossef et al. [2]. Motivating applications include satellite transmission of large files, audio and video on demand (such as streaming networks), database data retrieval, cache management for network applications and sensor networks. The model considered in [1] involves a source that possesses n messages and m receivers. Each receiver knows a proper subset of messages, which is referred to as the side information and demands a specific message unknown to it. The source, aware of the messages possessed by each receiver, uses this knowledge to develop a transmission scheme that satisfies the demands of all receivers using as few transmissions as possible, referred to as the index code length.

Index coding can be viewed as special case of rate distortion with multiple receivers, each with some side information about the source [3]. Index coding has received considerable attention recently, motivated by applications in multi-user broadcast scenarios, such as audio and video on demand, streaming networks, satellite communications and by its connection to network coding. In [4,5], the equivalence between network encoding and index encoding has been established. This research topic has been extended in other directions, such as pliable index coding [6], a variation of index coding in which we still consider a server and m clients with side information, but where the receivers are *flexible* and satisfied to receive any message that is not in their side information set; such flexibility can reduce the amount of communication, sometimes significantly. This has applications in music streaming services or internet searching, such as *content distribution networks* (CDNs) [7]; a CDN manages servers in multiple geographically distributed locations, stores copies of the web content (including documents, images, audio and others) in its servers and attempts to direct each user request to a CDN location that will provide the best user experience.

In this application, each receiver may be interested in receiving any message that it does not already possess as side information. Suppose that we are searching for the latest news and we already have some information. We are happy if we obtain any additional news that we do not have, with minimum delay. Here, we do not specify the news. On a music streaming service, users do not know which song will play next; they are usually only guaranteed that it will be one of a certain group and that it will not be repeated. In online advertising systems, customers do not require a specific advertisement to view; it is the distributor who chooses which one will be placed on customers' screens. The distributor may wish to avoid repeating the same advertisement for the same customer, as this can decrease customer satisfaction.

How much we can gain in terms of bandwidth and user satisfaction, if recommendation systems become bandwidth-aware and take into account not only the user preferences? Song and Fragouli [8] formulated this as a new problem in the context of index coding, where they relaxed the index coding requirements and considered the case where the customer is satisfied to receive any message that they do not already have, with satisfaction proportional to their preference for that message.

A promising research area that has recently emerged is in how to use index coding to improve the communication efficiency in distributed computing systems, especially for data shuffling in iterative computations [9,10]. Index coding has been proposed to increase the efficiency of data shuffling, which can form a major communication bottleneck for big data applications. In particular, pliable index coding can offer a more efficient framework for data shuffling, as it can better leverage the many possible shuffling choices to reduce the number of transmissions.

The index coding problem subject to transmission errors was initially considered by Dau et al. [11]. In this work, we establish a connection between index coding and error-correcting codes with multiple interpretations from the tree construction of nested cyclic codes proposed in [12]. The notion of multiple interpretation using nested codes [13] is as follows: multiple information packets are separately encoded via linear channel codes, and then combined by addition and transmitted as a single codeword, minimizing the number of channel uses and offering error protection. The resulting packet can be decoded and interpreted in different ways, yielding an increase in error correction capability, depending on the amount of side information available at each receiver.

Part of the content of this paper was presented in [14]. In the current version, evidence to verify our claims has been added, as well as some examples. The results in this paper are an extension of the results in [12,14].

The main contributions of this paper are as follows.

- We verify that, for cyclic codes, there will not always be an increase in error correction capability between different levels of the code tree. For this reason, we initially restrict the study to Reed–Solomon codes since they are maximum separable distance (MDS) codes, and provide an increase in Hamming distance at each level. This means that, under certain conditions, knowledge of side information can be interpreted as an increase in error correction capability.
- We propose a new variant for the index coding problem, which we call "index coding with multiple interpretations". We assume that receivers demand all the messages from the source and that the sender is unaware of the subset of messages already known by the receivers. The sender performs encoding such that any side information may be used by the decoder in order to increase its error correction capability. Moreover, if a receiver has no side information, the decoder considers the received word to belong to the highest rate code, associated with the root node of the tree.
- We also propose a solution to relax some constraints on how side information should occur at the receivers, using graph coloring associated with the pliable index coding problem.

2. Preliminaries

2.1. Notation and Definitions

For any positive integer n, we let $[n] := \{1, \ldots, n\}$. We write $\mathbb{F}_q$ to denote the finite field of size q, where q is a prime power, and use $\mathbb{F}_q{}^{n \times t}$ to denote the vector space of all $n \times t$ matrices over $\mathbb{F}_q$.

2.2. Review of Linear and Cyclic Codes

We now introduce the notation and briefly review some of the relevant properties of linear and cyclic codes based on [15,16]. The purpose of a code is to add extra check symbols to the data symbols so that errors may be found and corrected at the receiver. That is, a sequence of data symbols is represented by some longer sequence of symbols with enough redundancy to protect the data. In general, to design coding schemes for receivers with side information, we will consider collections of linear codes that are of length n over $\mathbb{F}_q$.

Structure of Linear Block Codes

Recall that under componentwise vector addition and componentwise scalar multiplication, the set of n-tuples of elements from $\mathbb{F}_q$ is the vector space called $\mathbb{F}_q^n$. For the vectors $u = (u_1, \ldots, u_n) \in \mathbb{F}_q^n$ and $v = (v_1, \ldots, v_n) \in \mathbb{F}_q^n$, the *Hamming* distance between u and v is defined to be the number of coordinates u and v that differ, i.e.,

$$d(u, v) = |\{i \in [n] : u_i \neq v_i\}|.$$

Definition 1. *A k-dimensional subspace $\mathcal{C}$ of $\mathbb{F}_q^n$ is called a linear $(n, k, d)_q$ code over $\mathbb{F}_q$ if the minimum distance of $\mathcal{C}$,*

$$d(\mathcal{C}) \triangleq \min_{u,v \in \mathcal{C},\, u \neq v} d(u, v)$$

is equal to d. Sometimes, we only use $(n, k)_q$ to refer to the code $\mathcal{C}$, where n is the length of the codewords and k is the dimension of the code. The code's rate is the ratio $\frac{k}{n}$.

That is, a $(n, k)_q$ linear code $\mathcal{C}$ can be completely described by any set of k linearly independent codewords $v_1, v_2, \ldots, v_k$; thus, any codeword is one of the q^k linear combinations $\sum_{i_1}^{k} \alpha_i v_i$, $\alpha_i \in \mathbb{F}_q$. If we arrange the codewords into a $k \times n$ matrix G, we say that G is a *generator matrix* for $\mathcal{C}$.

A special case of major importance is $\mathbb{F}_2^n$, which is the vector space of all binary codewords of length n with two such vectors added by modulo-2 addition in each component. A binary code of size $M = 2^k$ for an integer k is referred to as an (n, k) binary code.

We consider cyclic codes of length n over $\mathbb{F}_q$ with $gcd(n, q) = 1$. Label the coordinates of $c \in \mathbb{F}_q^n$ with the elements of $\mathbb{Z}_n = \{0, 1 \ldots, n-1\}$ and associate the vector $c = (c_0, \ldots, c_{n-1})$ with the polynomial $c(x) = c_0 + c_1 x + \cdots + c_{n-1} x^{n-1}$. With this correspondence, a cyclic code $\mathcal{C}$ is an ideal of the ring $R_n = \mathbb{F}_q[x]/(x^n - 1)$. We use $g(x)$ to denote the generator polynomial of $\mathcal{C}$ and write $\mathcal{C} = \langle g(x) \rangle = \{C(x) \in F_q[x];\ g(x) | C(x)\}$ to describe a t-error correcting cyclic code.

2.3. Index Coding with Side Information

The system shown in Figure 1 illustrates the index coding problem. Receiver R_i is requesting the message x_i, $i \in \{1, 2, 3\}$ and knows other messages as side information; R_1 knows x_3, R_2 knows x_1 and x_3 and the receiver R_3 knows x_1 and x_2.

The goal of index coding is to perform the joint encoding of the messages, in order to simultaneously meet the demands of all receivers, while transmitting the resulting messages at the highest possible rate.

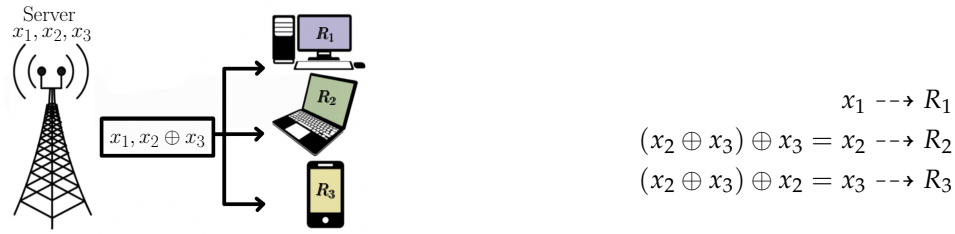


Figure 1. Index coding problem with three receivers.

Assuming a noiseless broadcast channel, the server would communicate all messages by sending one at a time, in three transmissions.

Alternatively, when transmitting the two coded messages x_1 and $x_2 \oplus x_3$, the receiver R_1 decodes x_1, from $(x_2 \oplus x_3) \oplus x_3 = x_2$ and $(x_2 \oplus x_3) \oplus x_2 = x_3$, R_2 and R_3 recover their demands.

The index coding problem is formulated as follows. Suppose that a server S wants to send a vector $x = (x_1, x_2, \ldots, x_n)$, where $x_i \in \mathbb{F}_q \; \forall \; i \in [n]$, to $[n]$ receivers $R_1, R_2, \ldots, R_n$. Each receiver R_i has $x_{\mathcal{S}_i} = \{x_j \; ; \; j \in \mathcal{S}_i \subseteq [n] \setminus \{i\}\}$ as side information and is interested in receiving the message x_i. The codeword $\mathfrak{C}(x) \in \mathbb{F}_q^\ell$ is sent and allows each receiver R_i to retrieve x_i. $\mathfrak{C}$ is an index code *scalar* over $\mathbb{F}_q$ of length ℓ. The purpose of S is to find an index code that has the minimum length. The index code is called linear if $\mathfrak{C}(x)$ is a linear function.

Index Coding via Fitting Matrices

A directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with n vertices specifies an instance of the index coding problem. Each vertex of $\mathcal{G}$ corresponds to a receiver (and its demand) and there is a directed edge $i \longrightarrow j$ if and only if the receiver R_i knows x_j as side information. Then, we write:

$$\mathcal{S}_i = \{ \; j \; : \; (i,j) \;\; \text{is a edge of} \;\; \mathcal{G} \; \}$$

Definition 2. *Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be a directed graph on n vertices without self-loops.*

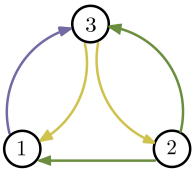
1. *A 0-1 matrix $M = (m_{ij})$, whose rows and columns are labeled by the elements of $\mathcal{V} = [n]$, fits $\mathcal{G}$ if, for all i and j,*
   (i) $m_{ii} = 1$;
   (ii) *For $i \neq j$,*
   $$m_{ij} = \begin{cases} * \in \{0,1\} \; ; \; \text{if } (i,j) \text{ is an edge of } \mathcal{G}; \\ 0 \qquad\qquad\quad ; \; \text{else.} \end{cases}$$
   Thus, $M - I$ is the adjacency matrix of an edge-induced subgraph of $\mathcal{G}$, where I denotes the $n \times n$ identity matrix.
2. *The minrank of $\mathcal{G}$ over the field $\mathbb{F}_2$ is defined as follows:*
   $$minrk_2(\mathcal{G}) \triangleq min\{ \; rank_2(M) \; : \; M \text{ fits } \mathcal{G} \; \}$$

Remark 1. *The term $rank_2(M)$ denotes the rank of such matrix M over $\mathbb{F}_2$, after " $*$ " has been assigned a value of 0 or 1. As an example for the index coding problem instance described in Figure 1, the matrix M would be given as follows:*

$$M = \begin{bmatrix} 1 & 0 & * \\ * & 1 & * \\ * & * & 1 \end{bmatrix}$$

Example 1. *Consider the side information graph $\mathcal{G}$ and a matrix M that fits $\mathcal{G}$, as shown in Figure 2. As $minrank_2(M) = 2$, we can select two linearly independent rows in a matrix M,*

namely M', and design an linear index code with the shortest possible length. The codeword sent will be $M'x$.



$$M = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 1 & 1 \end{bmatrix} \Rightarrow M'x = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} x_1 \\ x_2 \oplus x_3 \end{bmatrix}$$

Figure 2. Graph and matrix related to the problem described in Figure 1.

Theorem 1 ([2]). *For any side information graph $\mathcal{G}$, there exists a linear index code for $\mathcal{G}$ whose length equals $minrk_2(\mathcal{G})$. This bound is optimal for all linear index codes $\mathcal{G}$.*

In [17], the index encoding problem was generalized. Suppose that a sender wants to transmit n messages $(X_1, \ldots, X_n)$, where $X_i \in {\mathbb{F}_q}^t \ \forall\, i \in [n]$, to m receivers $R_1, \ldots, R_m$, through a noiseless channel. The receiver R_j is interested in recovering a single block $X_{f(j)}$, where $f : [m] \longrightarrow [n]$, and knows $X_{\mathcal{S}_j} = \{X_i \,;\, i \in \mathcal{S}_j \subseteq [n] \setminus f(j)\}$. The goal is to satisfy the demands of all receivers, exploiting their side information in a minimum number of transmissions.

When $m = n$, $f(j) = j$, $\forall\, j \in [m]$ and $t = 1$, we have a *scalar index code* [2]. Otherwise, we have a *vector index code*.

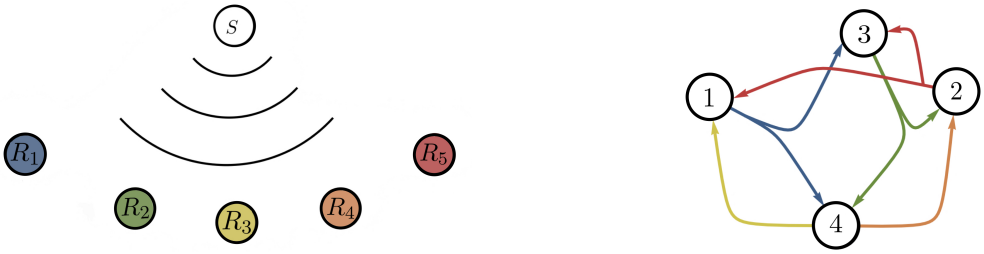
Let $\mathcal{I} = \{\mathcal{S}_j \,;\, j \in [m]\}$ be the set of side information of all receivers. An instance of an index coding problem given by $(m, n, \mathcal{I}, f)$ can be described by a directed hypergraph.

Definition 3. *The side information hypergraph $\mathcal{H}(\mathcal{V}, \mathcal{E}_\mathcal{H}) = \mathcal{H}(m, n, \mathcal{I}, f)$ is defined by the set of vertices $\mathcal{V} = [n]$ and the (directed) hyperedges $\mathcal{E}_\mathcal{H}$, where*

$$\mathcal{E}_\mathcal{H} = \{e_j = (f(j), \mathcal{S}_j) \,;\, j \in [m]\}$$

A hyperedge $e_j = (f(j), \mathcal{S}_j)$ represents the demand and side information of the receiver R_j.

Example 2. *Consider an instance of an index coding problem in Figure 3. The hypergraph in Figure 3b describes the problem, where $n = 4$ (messages) and $m = 5$ (receivers) requiring $f(1) = 1$, $f(2) = 3$, $f(3) = 4$, $f(4) = 4$ and $f(5) = 2$, and with the following side information sets $\mathcal{S}_1 = \{3, 4\}, \mathcal{S}_2 = \{2, 4\}, \mathcal{S}_3 = \{1\}$, $\mathcal{S}_4 = \{2\}$ and $\mathcal{S}_5 = \{1, 3\}$, respectively.*



(**a**) (**b**)

Figure 3. A single sender with multiple receivers having side information: (**a**) An instance of an index coding problem with $m = 5$ (receivers) and $n = 4$ (messages). (**b**) The hypergraph that describes this instance will have four vertices and five hyperedges: $e_1 = (1, \{3, 4\})$, $e_2 = (3, \{2, 4\})$, $e_3 = (4, \{1\})$, $e_4 = (4, \{2\})$ and $e_5 = (2, \{1, 3\})$.

Definition 4. *Given an instance of an index encoding problem described by $\mathcal{H}(m, n, \mathcal{I}, f)$,*

$$\mathfrak{C} : {\mathbb{F}_q}^{n \times t} \longrightarrow {\mathbb{F}_q}^{\ell \times t},$$

is a $\mathbb{F}_q$- index code with length ℓ, for the instance described by $\mathcal{H}$, if, for each receiver R_j, $j \in [m]$, there exists a decoding function

$$\mathfrak{D}_j : {\mathbb{F}_q}^{\ell \times t} \times {\mathbb{F}_q}^{t|\mathcal{S}_j|} \longrightarrow {\mathbb{F}_q}^{t},$$

satisfying $\mathfrak{D}_j(\mathfrak{C}(X), X_{\mathcal{S}_j}) = X_{f(j)}$, $\forall\ X \in\ {\mathbb{F}_q}^{n \times t}$.

The *transmission rate* of the code is defined as $\frac{\ell}{t}$. If $t = 1$, then the index code is known as a *scalar index code*; otherwise, it is known as a *vector index code*. A linear coding function $\mathfrak{C}$ is also called a linear index code. The goal of index coding is to find optimal index codes, i.e., those with the minimum possible transmission rate. For scalar linear index codes, we refer to the quantity r as the length of the code, and thus rate optimality translates to minimal length codes.

Definition 5. *$\mathfrak{C}$ is a $\mathbb{F}_q$-linear index code, $\mathfrak{C}(X) = GX$, $\forall\ X \in \mathbb{F}_q^{\ell \times n}$, where $G \in \mathbb{F}_q^{\ell \times n}$. G is the matrix that generates the linear index code $\mathfrak{C}$.*

The following definition generalizes the minrank definition over $\mathbb{F}_q$ of the side information graph $\mathcal{G}$, which was defined in [2], to a hypergraph $\mathcal{H}(m, n, \mathcal{I}, f)$.

Definition 6. *Let $Supp(v) = \{\ i\ \in\ [n] : v_i \neq 0\}$, the support of a vector $v \in \mathbb{F}_q^n$. The Hamming weight of v will be denoted by $\omega(v) = |Supp(v)|$, the number of nonzero coordinates of v.*

Definition 7 ([11])**.** *Suppose that $\mathcal{H}(m, n, \mathcal{I}, f)$ corresponds to an instance of index coding with side information (ICSI). Then, the minrank of $\mathcal{H}$ over $\mathbb{F}_q$ is defined as*

$$minrk_q(\mathcal{H}) \triangleq min\{\ rank_q(\{v_i + e_{f(i)}\}_{i \in [m]})\ :\ v_i \in \mathbb{F}_q\ Supp(v_i) \subseteq \mathcal{S}_i\}$$

This may be rewritten as follows.

Definition 8. *Let a side information hypergraph $\mathcal{H}$ correspond to an instance of the ICSI problem. A matrix $M = (m_{ij}) \in \mathbb{F}_q^{m \times n}$ fits $\mathcal{H}$ if*

$$m_{ij} = \begin{cases} 1 & if\ j = f(i) \\ 0 & if\ j \notin \mathcal{S}_i \end{cases}$$

The minrank of $\mathcal{H}$ over the field $\mathbb{F}_q$ is defined as follows:

$$minrk_q(\mathcal{H}) \triangleq min\{\ rank_q(M)\ :\ M\ fits\ \mathcal{H}\ \}$$

Theorem 2 ([2])**.** *Given an instance of an index encoding problem described by the hypergraph $\mathcal{H}(m, n, \mathcal{I}, f)$, the optimal length of an index code on the field $\mathbb{F}_q$ is $minrk_q(\mathcal{H})$.*

In [2], it was proven that, in several cases, linear index codes were optimal. They conjectured that for any side information graph $\mathcal{G}$, the shortest-length index code would always be linear and have length $\text{minrk}_2(\mathcal{G})$. The conjecture was refuted by Lubetzky and Stav in [18]. In any case, as shown by Peeters [19], calculating the minrank of an arbitrary graph is a difficult task. More specifically, Peeters showed that deciding whether a graph has minrank three is an NP-complete problem.

Example 3. *Consider the instance of the index encoding problem given in Example 2. Then, we find that the matrix M that fits the hypergraph $\mathcal{H}$ has the form:*

$$M = \begin{bmatrix} 1 & 0 & * & * \\ 0 & * & 1 & * \\ * & 0 & 0 & 1 \\ 0 & * & 0 & 1 \\ * & 1 & * & 0 \end{bmatrix}$$

The lines are associated with the receivers $R_1, \ldots, R_5$ and the columns to the message indexes $1, 2, 3$ and 4. The symbol " $*$ " can be replaced by an arbitrary element in the field $\mathbb{F}_q$.

For an example, consider the field $\mathbb{F}_2$. A matrix that fits the hypergraph $\mathcal{H}$ has rank at least 3. Thus, we select

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

which achieves the minimum rank 3. Now, we consider three linearly independent lines of M, and suppose that

$$G = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \Rightarrow \quad Gx = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} x_1 \\ x_2 \oplus x_3 \\ x_4 \end{bmatrix}$$

The decoding process goes as follows. Since R_2 and R_5 already know $\{x_2, x_4\}$ and $\{x_1, x_3\}$, respectively, they obtain x_3 and x_2, respectively, from the first packet. Receiver R_1 obtains x_1 and both R_3 and R_4 obtain x_4.

Remark 2. *We have made available at [20] an algorithm (m-files) in Matlab, which is designed to solve small examples in this work, since, as we mentioned above, there is no polynomial-time algorithm for an arbitrary graph.*

2.4. Pliable Index Coding

The pliable index coding problem (PICOD), introduced by Brahma and Fragouli in [6], is a variant of the index coding problem. In PICOD, users do not have predetermined messages to decode, as in the case of classic index coding; instead, each user is satisfied to decode any message that is not present in its side information set. Figure 4 illustrates this system model.

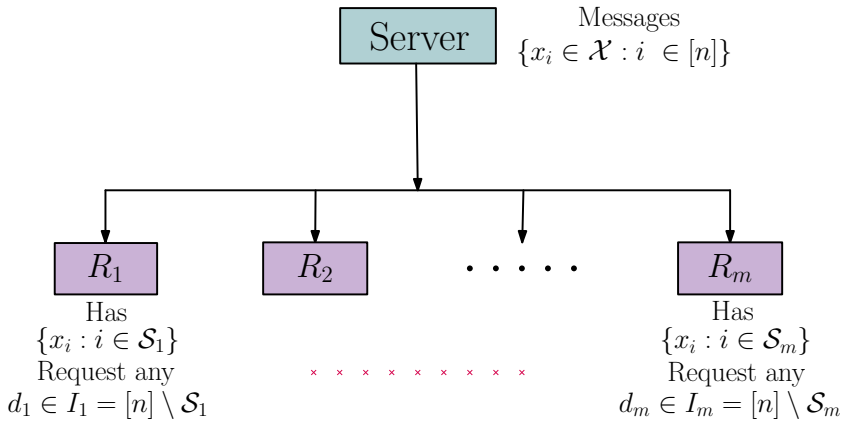


Figure 4. Pliable index coding scheme.

The problem is formalized as follows: a transmitter with n messages $\{x_i : i \in [n]\}$, $x_i \in \mathcal{X}$ is connected to m receivers $R_1, \ldots, R_m$, through a noiseless channel. Each receiver R_j knows $x_{\mathcal{S}_j} = \{x_i : i \in \mathcal{S}_j\}$ as side information. We denote by $I_j \triangleq [n] \setminus \mathcal{S}_j$ the index set of the unavailable messages in $x_{\mathcal{S}_j}$. Then, $x_{I_j} = \{x_i : i \in I_j\}$ denotes the set of requests of R_j. Each receiver R_j is satisfied if it can successfully recover any message that is not present in its side information set, i.e., any message $x_d \in x_{I_j}$.

We can represent an instance of a pliable index coding problem using an undirected bipartite graph, one side representing the message indexes and the other side representing the receivers. We connect R_j to the indices belonging to I_j, as in Figure 5.

Remark 3. *By having this freedom to choose the desired message for each user, PICOD can satisfy all users with a significant reduction in the number of transmissions compared to the index encoding problem with the same set of messages and the same sets of user side information.*

Example 4. *We will consider the case described in Example 2 as a pliable index coding problem. Now, we have the bipartite graph in Figure 5 describing the problem. Note, for example, that client 1 demands any of the messages indexed in $I_1 = \{1, 2\}$ and knows the indexed messages in $\mathcal{S}_1 = \{3, 4\}$; client 3 will be satisfied to receive any of the messages x_2, x_3 or x_4, since it only knows x_1.*

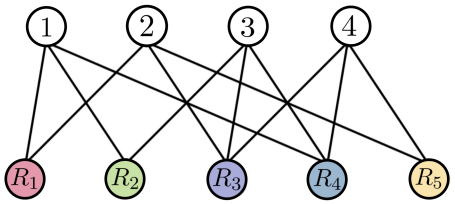


Figure 5. Bipartite graph for PICOD.

Pliable Index Coding via Colorings of Hypergraphs

In [21], a graph coloring approach was presented for pliable index coding. The authors have shown the existence of a coding scheme that has length $O(\log^2 \Gamma)$, where Γ refers to a hypergraph parameter that captures the maximum number of intersections between edges of the PICOD hypergraph.

An instance of the pliable index encoding problem is described by $(m, n, \mathcal{I})-$PICOD, onde $\mathcal{I} = \{I_j ; j \in [m]\}$, and can be conveniently represented by a hypergraph.

Definition 9. *The hypergraph $\mathcal{H}(\mathcal{V}, \mathcal{E}_\mathcal{H})$, with $\mathcal{V} = [n]$ vertices and $\mathcal{E}_\mathcal{H} = \{e_j = (I_j) ; j \in [m]\}$ hyperedges, completely describes the $(m, n, \mathcal{I})-$PICOD. The hyperedge $e_j = (I_j)$ represents the set of requests for R_j (i.e., $\mathcal{E}_\mathcal{H} = \mathcal{I}$).*

The problem illustrated in Example 5 can be represented by a hypergraph, as can be seen in Figure 6.

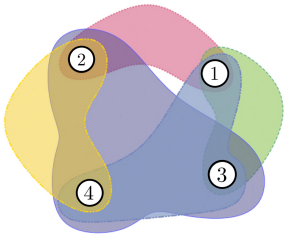


Figure 6. Hypergraph.

Let $\mathcal{H} = (\mathcal{V}, \mathcal{E}_\mathcal{H})$ be a hypergraph and $C : \mathcal{V} \to [L]$ be a coloring of $\mathcal{V}$, where L is a positive integer. We say that C is a conflict-free coloring for the hyperdges, if each $\mathcal{E}_\mathcal{H}$ of $\mathcal{H}$ has at least one vertex with unique color. The smallest number of colors required for such a coloring is called the conflict-free chromatic number of $\mathcal{H}$, denoted by $\chi_{CF}(\mathcal{H})$. This parameter was first introduced by Even et al. [22].

Remark 4. *In [21], pliable index coding was given a graph coloring approach. The authors have shown the existence of a coding scheme that has length $O(\log^2 \Gamma)$, where Γ refers to a hypergraph parameter that captures the maximum number of "incidences" of other hyperedges in any given hyperedge. This result improves the best known achievable results, in some parameter regimes.*

Definition 10. *A pliable index code (PIC) consists of a collection of an encoding function on the server that encodes the n messages to an $\ell-$length codeword,*

$$\phi : \mathcal{X}^n \longrightarrow \mathcal{X}^\ell,$$

and decoding functions $\psi_j : \mathcal{X}^\ell \times \mathcal{X}^{|\mathcal{S}_j|} \longrightarrow \mathcal{X}$, satisfying $\psi_j(\phi(x), x_{\mathcal{S}_j}) = x_d$, for some $d \in I_j$.

The quantity $\ell-$ is called the length of the pliable index code. We are interested in designing pliable index codes that have small $\ell-$.

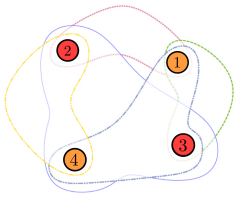
We will assume that $\mathcal{X} = \mathbb{F}^k$ for some finite field $\mathbb{F}$ and integer $k \geq 1$. If $k > 1$, we refer to this code as a $k-$*vector* PIC, while the $k = 1$ case is also called a *scalar* PIC. We will concentrate on the linear PICs. In this case, the coding function ϕ is represented by a matrix $\ell \times mk$ (denoted by G) such that $\phi(x_i : i \in [n]) = Gx^T$, where $x = (x_{11}, \ldots, x_{1k}, \ldots, x_{m1}, \ldots, x_{mk})$. The smallest ℓ for which there is a linear $k-$*vector* PIC for an instance of the pliable index coding problem given by the hypergraph $\mathcal{H}$ will be denoted by $\ell_k^*(\mathcal{H})$.

Definition 11. *Let $C : \mathcal{V} \to [L]$ be a conflict-free coloring of the hypergraph $\mathcal{H}$ that represents a PICOD. The indicator matrix associated with this coloring G_C, $L \times n$, is given by*

$$G_c(c, i) = \begin{cases} 1, & \text{if the vertex } i \text{ received the color } c; \\ 0, & \text{otherwise.} \end{cases}$$

Teorema 1 ([21]). *The indicator matrix G_c generates the pliable index code for the problem described by the hypergraph $\mathcal{H}$.*

Example 5. *Consider the same PICOD represented in Figure 6. The coloring shown in Figure 7 is a conflict-free coloring with two colors. Then, the matrix $G_c = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix}$.*



$$\Rightarrow \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} x_1 \oplus x_4 \\ x_2 \oplus x_3 \end{bmatrix}$$

Figure 7. Conflict-free coloring with two colors.

From the messages $x_1 \oplus x_4$ and $x_2 \oplus x_3$, all receivers can successfully recover at least one message from their request set.

Using the same parameters as in Example 2, we see that the length of the index code for this instance is $\ell = 3$, while, for the PICOD case, $\ell_k^*(\mathcal{H}) = 2$.

2.5. Index Coding via MDS Codes

The index coding model defined in Section 2.3, via graph theory, is only one of many approaches used to describe and solve an index encoding problem. One of the most interesting index coding schemes using codes has the maximum distance separable (MDS) property, which consists of transmitting $\kappa(\mathcal{G}) = n - \text{mindeg}(\mathcal{G})$, the parity symbols of a systematic MDS code with parameters $(n + \kappa, n)$, where $\text{mindeg}(\mathcal{G})$ represents the minimum amount of side information available at each receiver, i.e., for a general index encoding problem with side information graph $\mathcal{G}$,

$$\text{mindeg}(\mathcal{G}) \triangleq \min_{i\in[n]} |\{ j; (i,j) \in \mathcal{E}(\mathcal{G})\}| = \min_{i\in[n]} |\mathcal{S}_i|.$$

Then, every receiver has n code symbols (including its side information) and, by the MDS property, it can successfully recover its desired message.

Proposition 1 ([1]). *Consider an index coding problem with n messages and n receivers represented by the side information graph $\mathcal{G}$. Let $\mathcal{S}_i$, the side information set of the receiver $R_i, i \in [n]$, and then*

$$\text{minrk}_q(\mathcal{G}) \leq \kappa(\mathcal{G}) = n - \text{mindeg}(\mathcal{G}) = n - \min_{i\in[n]} |\mathcal{S}_i|.$$

Corollary 1. *If $\mathcal{G}$ is a complete graph, then $mindeg(\mathcal{G}) = n - 1$ and the transmission of the parity symbol $x_1 + \cdots + x_n$ of an $(n + 1, n)$ MDS code over $\mathbb{F}_2$ achieves $minrk_q(\mathcal{G}) = 1$.*

3. Results

The tree construction method proposed in [12] can be interpreted as a network coding problem with multiple sources and multiple users. In the proposed model, both encoding and decoding are performed by polynomial operations, without the need for side information; however, if they exist, they may allow multiple interpretations at the receivers, based on the side information available at each receiver. Figure 8 illustrates this system model.

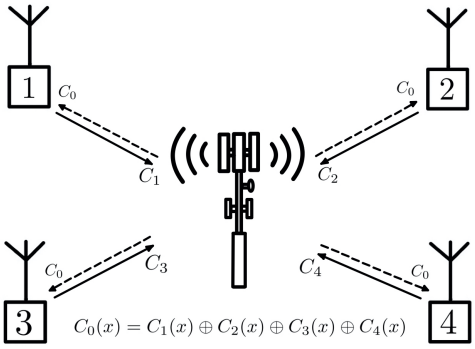


Figure 8. Coding model with nested cyclic codes.

Given the connection between network and index coding problems, established in [4], we can also interpret the coding with nested cyclic codes, at the stage where the packets are XOR-ed together, as a case of MDS index coding according to Corollary 1, in the particular case where each receiver is unaware of the message it is requesting, which may be a rare occurrence. However, it is possible to take advantage of the method's distinguishing feature—the possibility of multiple interpretations at the receivers—and, by imposing some extra conditions, design an index code model that has greater flexibility over the side information sets.

In the next subsections, we present some results and algorithm implementations, and in Section 4, we present in detail the proposed index encoding with multiple interpretations.

3.1. Index Coding from Reed–Solomon Codes

We establish a connection between index coding and error-correcting codes based on the tree construction method of nested cyclic codes proposed in [12]. We implement a few algorithms to perform tree construction using the Matlab language, which allows us to work over finite bodies in a practical and efficient way and helps to solve some implementation problems encountered later in [12]. We prove that for cyclic codes, there will not always be an increase in error correction capability between the levels of the tree, as suggested in [12]. This is why we have initially limited this study to Reed–Solomon codes, because they are MDS codes, which guarantees an increase in Hamming distance at each level, meaning that, under certain conditions, the knowledge of side information will be interpreted as an increase in the decoder's ability to correct errors.

A Tree Construction with Nested Cyclic Codes

A nested code is characterized by a global code where each element is given by a sum of codewords, each belonging to a different subcode. That is,

$$c = i_1 G_1 \oplus i_2 G_2 \oplus \cdots \oplus i_N G_N,$$

where $\oplus$ represents an XOR operation. For an information vector i_ℓ, $1 \leq \ell \leq N$, the codeword $i_\ell G_\ell$ belongs to a subcode $\mathcal{C}_\ell$ of code $\mathcal{C}$ and $c \in \mathcal{C}$.

Nested cyclic codes, whose subcodes are generated by generator polynomials, were originally proposed by Heegard [23], and were originally called partitioned linear block codes. They can be defined as follows:

Definition 12. *Let $\mathcal{C} = \{C(x) \in F_q[x];\ g(x)|C(x)\}$ be a t-error-correcting cyclic code having $g(x)$ as the generator polynomial. Note that $\mathcal{C} = \langle g(x) \rangle$ is an ideal of the ring $R_n = \mathbb{F}_q[x]/(x^n - 1)$, but is also a vector subspace of $\mathbb{F}_q^n$, such that*

$$C(x) = p_1(x)g_1(x) + p_2(x)g_2(x) + \cdots + p_N(x)g_N(x),$$

where $C_\ell(x) = p_\ell(x)g_\ell(x)$, $1 \leq \ell \leq N$ is an encoded packet belonging to the t_ℓ-error-correcting subcode

$$\mathcal{C}_\ell = \{C_\ell(x) \in F_q[x];\ g_\ell(x)|C_\ell(x)\},$$

generated by $g_\ell(x)$ and satisfying the following conditions:

1. $g_\ell(x)|g_{\ell+1}(x)$;
2. $\deg[C_\ell(x)] < \deg[g_{\ell+1}(x)]$.

The tree-based algebraic construction of nested cyclic codes, proposed in [12], aims to

1. Encode, independently, different data packets, providing protection against channel errors;
2. Encode different data packets producing codewords that are added, resulting in the packet C_0;
3. Correct the errors on C_0 and, finally, recover the data in the receiver by polynomial operations.

Consider a tree in which the root node is associated with the vector subspace of an encompassing error correcting code. Thus, the root node is defined as the code $\mathcal{C}_{i0}$, such that

$$\mathcal{C}_{i0} = \langle g_{i0}(x) \rangle = \{C_{i0}(x) \in F_q[x];\ g_{i0}(x)|C_{i0}(x)\}.$$

This subspace corresponds to a t_0-error-correcting cyclic code $\mathcal{C}_{i0}(n, k_{i0})$, generated by the polynomial $g_{i0}(x)$.

Definition 13. *A tree of nested cyclic codes is a finite tree such that*

1. *Each inner node (including the root node) can be subdivided into another inner node and a terminal node;*

2. *The jth th inner node is associated with a linear subspace $\mathcal{C}_{ij} \subset F_q^n$ of dimension k_{ij}, and can be subdivided into the subspaces*

$$\mathcal{C}_{ij} = \mathcal{C}_{i(j+1)} + \mathcal{C}_{t(j+1)} \ \ com \ \ \mathcal{C}_{i(j+1)} \cap \mathcal{C}_{t(j+1)} = \{0\} \ \ e \ \ k_{ij} = k_{i(j+1)} + k_{t(j+1)}$$

3. *The subspace $\mathcal{C}_{ij}$, associated with the jth inner node, must be a cyclic linear block code generated by $g_{ij}(x)$;*
4. *If $\mathcal{C}_{ij} = \langle g_{ij}(x) \rangle$ e $\mathcal{C}_{i(j+1)} = \langle g_{i(j+1)}(x) \rangle$, then $g_{ij}(x) | g_{i(j+1)}(x)$; furthermore, $g_{ij}(x) | x^n - 1$ for any $g_{ij}(x)$;*
5. *To conclude, the last inner node will have no ramifications.*

Remark 5. *Figure 9 illustrates the model described above.*

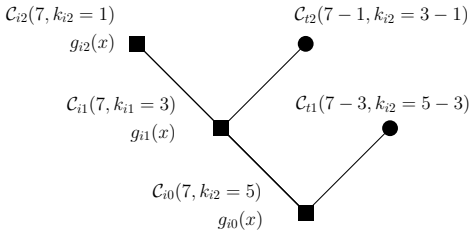


Figure 9. Tree construction. The sum of the dimensions associated with the last node and the terminal nodes is equal to the dimension of the root node.

Let $p_j(x)$ be the data packet associated with the terminal node, for $1 \leq j \leq T$. The encoding is given by

$$C_j(x) = p_j(x) g_{i(j-1)}(x).$$

Then, the encoded packets are summed up and the resulting codeword is sent out by the transmitter

$$C_0(x) = C_1(x) + C_2(x) + \cdots + C_T(x).$$

After the error correction phase, the jth packet $p_j(x)$ is decoded by the operations:

$$p_j(x) = \begin{cases} [C_0(x) \mod g_{ij}(x)] / g_{i(j-1)}(x) & \text{if } 1 \leq j \leq T, \\ C_0(x) / g_{i(T-1)}(x) & \text{if } j = T. \end{cases} \quad (1)$$

The information will be contained in the remainder of the division of $C_0(x)$ by $g_{ij}(x)$, since the modulo operation eliminates the influence of all messages related to polynomials of degree equal to or greater than the degree of g_{ij}. Thus, the quotient of the final division operation provides the desired information, since all other messages have degree less than the degree of the divisor polynomial. Therefore, in the case of the last package, only the division operation is required. We suggest consulting [12] for more details on the encoding process using the tree construction method.

3.2. Tree Construction: Algorithm and Considerations

We describe a few algorithms in Matlab and considerations for fitting to the model of tree construction, which can be found at [20], allowing us to perform the calculations on finite fields by making the appropriate transformations from integer representation to powers of α. Below, we exemplify the main idea of the algorithm.

Example 6. *For $T = 3$ let $\mathcal{C}_{i0}(7,5)$ be a Reed–Solomon code in $GF(8)$ and $k_{t1} = k_{t2} = 2$ the dimensions of subspaces $\mathcal{C}_{t1}, \mathcal{C}_{t2}$, respectively. They are associated with the terminal nodes of the tree; the last node of the tree, which is an inner node without ramification, is associated with $\mathcal{C}_{i2}$ of dimension $k_{i2} = 1$.*

The packets $p_1(x) = x + \alpha^2$, $p_2(x) = \alpha^3 x + \alpha$, both associated with the the terminal nodes, have length 2; $p_3(x) = \alpha^5$ has length equal to 1 and is associated with the last node. Let α be the primitive element of $GF(8)$, and the generator polynomials are

1. $\deg(g_{i0}(x)) = n - k_{i0} = 2 \Rightarrow \; g_{i0}(x) = \prod_{j=1}^{2}(x - \alpha^j) = x^2 + \alpha^4 x + \alpha^3;$
2. $\deg(g_{i1}(x)) = n - k_{i1} = 4 \Rightarrow \; g_{i1}(x) = \prod_{j=1}^{4}(x - \alpha^j) = x^4 + \alpha^3 x^3 + x^2 + \alpha x + \alpha^3;$
3. $\deg(g_{i2}(x)) = n - k_{i2} = 6 \Rightarrow \; g_{i2}(x) = \prod_{j=1}^{6}(x - \alpha^j) = x^6 + x^5 + x^4 + x^3 + x^2 + x + 1.$

Then, the encoded packets are

$$\begin{aligned} C_1(x) &= p_1(x)g_{i0}(x) \\ &= x^3 + \alpha x^2 + \alpha^4 x + \alpha^5; \\ C_2(x) &= p_2(x)g_{i1}(x) \\ &= \alpha^3 x^5 + \alpha^5 x^4 + \alpha^6 x^3 + \alpha^2 x^2 + x + \alpha^4; \\ C_3(x) &= p_3(x)g_{i2}(x) \\ &= \alpha^5 x^6 + \alpha^5 x^5 + \alpha^5 x^4 + \alpha^5 x^3 + \alpha^5 x^2 + \alpha^5 x + \alpha^5. \end{aligned}$$

The transmitted codeword $C_0(x)$ is given by

$$\begin{aligned} C_0(x) &= C_1(x) + C_2(x) + C_3(x) \\ &= \alpha^5 x^6 + \alpha^2 x^5 + 0x^4 + \alpha^3 x^3 + 1x^2 + 0x + \alpha^4. \end{aligned}$$

Remark 6. *Each terminal node is a shortened version of the code associated with the inner node from which the terminal node emanates. It is implicit that the codewords of shortened codes are prefixed with zeros to achieve length n and, therefore, that these codes are not cyclic.*

3.2.1. Decoding—Error Correction

Considering tree construction based on Reed–Solomon codes and assuming that the receiver has side information available, when will there be an increase in error correction capability?

Proposition 2. *Due to the nesting structure, the variable error correctability characteristic can only be observed if there is a sequential removal of the packets associated with the nodes from the root to the top of the tree.*

Proof. Supposing that $C_\ell(x)$, $1 \leq \ell \leq T$ is the first coded packet known at the receiver, then

$$\begin{aligned} C_0(x) &= p_1(x)g_{i0}(x) + \cdots + p_{(\ell-1)}(x)g_{i(\ell-2)}(x) + p_{(\ell+1)}(x)g_{i\ell}(x) + \cdots + p_T(x)g_{i(T-1)}(x) \\ &= [p_1(x) + \cdots + p_{(\ell-1)}(x)q_{(\ell-1)}(x) + p_{(\ell+1)}(x)q_{(\ell+1)}(x) + \cdots + p_T(x)q_T(x)]g_{i0}(x), \end{aligned}$$

therefore, $C_0(x) \in \mathcal{C}_{i0}(n, k_{io})$, whose error correction capability is t_0. Note that even though the receiver knows about other packages $C_j(x)$, $\ell < j \leq T$, the result does not change. On the other hand, if all packages $C_j(x)$; $1 \leq j < \ell$ are known to the receiver, we can write

$$\begin{aligned} C_0(x) &= p_{(\ell+1)}(x)g_{i\ell}(x) + \cdots + p_T(x)g_{i(T-1)}(x) \\ &= [p_{(\ell+1)}(x)\overline{q}_{(\ell+1)}(x) + \cdots + p_T(x)\overline{q}_T(x)]g_{i\ell}(x), \end{aligned}$$

thus, $C_0(x) \in \mathcal{C}_{i\ell}(n, k_{i\ell})$, whose error correction capability is $t_\ell \geqslant t_0$, and equality occurs only when

$$d_{min}(C_\ell) - d_{min}(C_0) < 2.$$

□

Example 7. *Consider the same tree as in Example 6.*

- *If all packages are unknown ⇒ the decoding is performed by* $\mathcal{C}_{i0}(7,5) \therefore t_0 = 1$*;*
- *If* $C_1(x)$ *is known ⇒ the decoding is performed by* $\mathcal{C}_{i1}(7,3) \therefore t_1 = 2$*;*
- *If* $C_2(x)$ *is known* $C_1(x)$ *⇒ the decoding is performed by* $\mathcal{C}_{i2}(7,1) \therefore t_2 = 3$*.*

However, if $C_2(x)$ *is known but* $C_1(x)$ *is not, then the resulting codeword still belongs to* $C_0(x) \in \mathcal{C}_{i0}(7,5)$*, and there is no improvement in error correction capability, since*

$$\begin{aligned} C_0(x) &= p_1(x)g_{i0}(x) + p_3(x)g_{i0}(x)\overline{q}(x) \\ &= [p_1(x) + p_3(x)\overline{q}(x)]g_{i0}(x). \end{aligned}$$

Another advantage of Reed–Solomon codes is that they are easily decoded using an algebraic method known as syndrome decoding.

Syndrome Decoding

Syndrome decoding is an algebraic method based on the Berlekamp–Massey algorithm, which became a prototype for the decoding of many other linear codes.

If the coded package $C_1(x)$ is known and an error $e(x)$ occurs, then the message received will be

$$r(x) = C_0(x) + C_1(x) + e(x)$$

Suppose that the error is given by $e(x) = 0x^6 + 0x^5 + \alpha^2x^4 + \alpha^5x^3 + 0x^2 + 0x + 0$, and then

$$r(x) = \alpha^5x^6 + \alpha^2x^5 + \alpha^2x^4 + \alpha^6x^3 + \alpha^3x^2 + \alpha^4x + 1.$$

Remark 7. *Notice that we need to find the error locations and their values, which is the main difference with binary codes, since, for binary codes, it is enough to determine the error locations.*

The decoding process can be divided into three stages.

1. Syndrome calculation

The syndrome calculation stage consists of checking the roots of the generating polynomial as inputs of $r(x)$. If the result is null, the sequence belongs to the set of codewords and, therefore, there are no errors. Any nonzero value indicates the presence of an error.

If the encoded packet $C_1(x)$ is known, then the error correction algorithm is executed by $\mathcal{C}_{i1}$, which is a $RS(7,3)$ code, generated by $g_{i1}(x) = (x-\alpha)(x-\alpha^2)(x-\alpha^3)(x-\alpha^4)$. Then,

$$r(x) = m(x)g_{i1}(x) + e(x),$$

Therefore, evaluating the roots of $g(x)$ at $r(x)$, the result will only be null when there are no errors in the transmission.

$$\therefore \quad S_i = r(x)\Big|_{x=\alpha^i} = e(\alpha^i), \quad \forall \quad i = 1, \ldots, n-k = 2t.$$

$$\begin{aligned} S_1 &= r(\alpha) = \alpha^5; & S_3 &= r(\alpha^3) = 0; \\ S_2 &= r(\alpha^2) = \alpha^6; & S_4 &= r(\alpha^4) = \alpha^6. \end{aligned}$$

2. Error Localization

Let μ be the number of errors $0 \leq \mu \leq t$, which occur at locations $\ell_1, \cdots, \ell_\mu$, and the error polynomial can be written as

$$e(x) = e_{\ell_1}x^{\ell_1} + \cdots + e_{\ell_\mu}x^{\ell_\mu}.$$

To correct $r(x)$, we must find the values and locations of the errors, which are denoted, respectively, by $e_{\ell_1}, \ldots, e_{\ell_\mu}$ and $x^{\ell_1}, \ldots, x^{\ell_\mu}$. Substituting $\alpha^j, 1 \leq j \leq 2t$, into the error polynomial $e(x)$, we obtain

$$\begin{cases} S_1 = e(\alpha) = e_{\ell_1}\alpha^{\ell_1} + e_{\ell_2}\alpha^{\ell_2} + \cdots + e_{\ell_\mu}\alpha^{\ell_\mu} \\ S_2 = e(\alpha^2) = e_{\ell_1}(\alpha^{\ell_1})^2 + e_{\ell_2}(\alpha^{\ell_2})^2 + \cdots + e_{\ell_\mu}(\alpha^{\ell_\mu})^2 \\ \vdots \\ S_{2t} = e(\alpha^{2t}) = e_{\ell_1}(\alpha^{\ell_1})^{2t} + e_{\ell_2}(\alpha^{\ell_2})^{2t} + \cdots + e_{\ell_\mu}(\alpha^{\ell_\mu})^{2t} \end{cases}.$$

Obtain $X_i = \alpha^{\ell_i}$ and $Y_i = e_{\ell_i}$ for $1 \leq i \leq \mu$, where X_i and Y_i will represent, respectively, the locations and values of the errors. Note that we will have $2t$ equations and $2t$ unknowns, t being error values and t being locations.

$$\begin{cases} S_1 = Y_1X_1 + Y_2X_2 + \cdots + Y_\mu X_\mu \\ S_2 = Y_1{X_1}^2 + Y_2{X_2}^2 + \cdots + Y_\mu X_\mu^2 \\ \vdots \\ S_{2t} = Y_1{X_1}^{2t} + Y_2{X_2}^{2t} + \cdots + Y_\mu X_\mu^{2t} \end{cases}.$$

It can be shown that this nonlinear system has a unique solution if $0 \leq \mu \leq t$ [15]. The techniques that solve this system of equations include defining the error locator polynomial (ELP) $\sigma(z)$ [24].

Definition 14. *Define the error locator polynomial sigma*(z)*, as*

$$\begin{aligned} \sigma(z) &= (1 - X_1z)(1 - X_2z)\cdots(1 - X_\mu z) \\ &= \sigma_\mu z^\mu + \cdots + \sigma_2 z^2 + \sigma_1 z + 1. \end{aligned}$$

The inverse of the square root of $\sigma(z)$*,* $1/X_1, \ldots, 1/X_\mu$*, indicates the locations of errors.*

To find error locations X_i, $1 \leq i \leq \mu$, note that $\sigma_1, \sigma_2, \ldots, \sigma_\mu$ and calculate the zeros of $\sigma(z)$; to find them, we use a syndrome matrix, as we see below:

$$\begin{bmatrix} S_1 & S_2 & \cdots & S_\mu \\ S_2 & S_3 & \cdots & S_{\mu+1} \\ \vdots & \vdots & \ddots & \vdots \\ S_\mu & S_{\mu+1} & \cdots & S_{2\mu-1} \end{bmatrix} \begin{bmatrix} \sigma_\mu \\ \sigma_{\mu-1} \\ \vdots \\ \sigma_1 \end{bmatrix} = \begin{bmatrix} -S_{\mu+1} \\ -S_{\mu+2} \\ \vdots \\ -S_{2\mu} \end{bmatrix}$$

Returning to Code RS(7,3), where the error correction capability is $t = 2$, we must find σ_1 e σ_2:

$$\begin{bmatrix} S_1 & S_2 \\ S_2 & S_3 \end{bmatrix} \begin{bmatrix} \sigma_2 \\ \sigma_1 \end{bmatrix} = \begin{bmatrix} S_3 \\ S_4 \end{bmatrix}$$
$$\begin{bmatrix} \alpha^5 & \alpha^6 \\ \alpha^5 & 0 \end{bmatrix} \begin{bmatrix} \sigma_2 \\ \sigma_1 \end{bmatrix} = \begin{bmatrix} 0 \\ \alpha^6 \end{bmatrix} \Rightarrow \sigma_2 = 1 \text{ e } \sigma_1 = \alpha^6$$

Thus, $\sigma(z) = z^2 + \alpha^6 z + 1$ with roots α^3 and α^4, so there is an error at the locations $\alpha^{-3} = \alpha^4$ and $\alpha^{-4} = \alpha^3$. Then,

$$e(x) = e_3x^3 + e_4x^4.$$

3. Determining the error values

Calculating $e(x) = e_3x^3 + e_4x^4$ at the points α and α^2, we can use the syndromes already obtained, S_1 and S_2, to determine the values of the errors, solving the following system:

$$\begin{cases} S_1 = e(\alpha) = e_3\alpha^3 + e_4\alpha^4 \\ S_2 = e(\alpha^2) = e_3\alpha^6 + e_4\alpha^8 \end{cases}$$

Therefore, the error polynomial is given by $e(x) = \alpha^5x^3 + \alpha^2x^4$. Now, correcting the received word $r(x)$, we have

$$\begin{aligned} C_0(x) &= r(x) + c_1(x) + e(x) \\ &= \alpha^5x^6 + \alpha^2x^5 + 0x^4 + \alpha^3x^3 + 1x^2 + 0x + \alpha^4. \end{aligned}$$

3.2.2. Decoding—Data Recovery

Example 8. *For the cases in the previous examples, where $T = 3$, the original data can be recovered as follows:*

- $p_1(x) = \dfrac{C_0(x) \mod g_{i1}(x)}{g_{i0}(x)} = \dfrac{p_1g_{i0} \mod g_{i1}}{g_{i0}(x)} = \dfrac{p_1(x)g_{i0}(x)}{g_{i0}(x)}$;
- $p_2(x) = \dfrac{C_0(x) \mod g_{i2}(x)}{g_{i1}(x)} = \dfrac{p_1g_{i0} \mod g_{i2} + p_2g_{i1} \mod g_{i2}}{g_{i1}(x)}$;
- $p_3(x) = \dfrac{C_0(x)}{g_{i2}(x)} = \dfrac{p_1(x)g_{i0}(x)}{g_{i2}(x)} + \dfrac{p_2(x)g_{i1}(x)}{g_{i2}(x)} + \dfrac{p_3(x)g_{i2}(x)}{g_{i2}(x)}$;

In summary, the module operation removes the branches above the node of interest and the division operation removes the branches below. Therefore, no side information is needed at the receiver in order to recover the data packets.

Will There Always Be an Increase in Error Correction Capability?

We analyze two cases of tree construction of nested cyclic codes, with the same parameters at each level. In one of them, we observe no increase in the error correction capability from the second to last internal node of the tree. This is due to the variety of possibilities of generating polynomials for a cyclic code of parameters (n, k). As a result, we demonstrate in Proposition 3 that, for Reed–Solomon codes, this feature of increasing capacity will be guaranteed provided that: $k_{ij} - k_{i(j+1)} \geq 2, \forall\, j = 0, \ldots, T-1$.

Example 9. *Let $\mathcal{C}_{i0}(15, 10)$ be a cyclic code in $GF(2)$ and $k_{t1} = 4, k_{t2} = 2$ be the dimensions of the subspaces $\mathcal{C}_{t1}, \mathcal{C}_{t2}$, respectively. The last node is associated with $\mathcal{C}_{i2}$ with dimension $k_{i2} = 4$. The construction is depicted in Figure 10.*

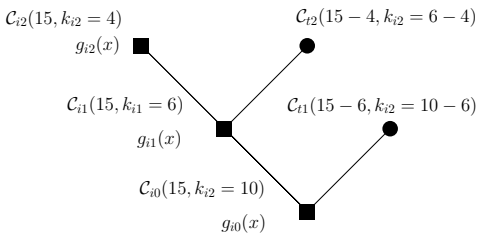


Figure 10. Tree construction.

We consider the factorization:

$$x^{15} - 1 = (1 + x)(1 + x^3 + x^4)(1 + x + x^2 + x^3 + x^4)(1 + x + x^2)(1 + x + x^4)$$

Case 1.

- $g_{i0}(x) = (1+x)(1+x^3+x^4) \quad \Rightarrow \quad t_0 = 1;$
- $g_{i1}(x) = g_{i0}(x)(1+x+x^2+x^3+x^4) \quad \Rightarrow \quad t_1 = 2;$
- $g_{i2}(x) = g_{i1}(x)(1+x+x^2) \quad \Rightarrow \quad t_2 = 3.$

Case 2.

- $g_{i0}(x) = (1+x)(1+x+x^4) \quad \Rightarrow \quad t_0 = 1;$
- $g_{i1}(x) = g_{i0}(x)(1+x^3+x^4) \quad \Rightarrow \quad t_1 = 2;$
- $g_{i2}(x) = g_{i1}(x)(1+x+x^2) \quad \Rightarrow \quad t_2 = 2.$

Remark 8. *We have provided an m-file algorithm at [20], which can be run through Matlab and performs the operations described in Examples 8 and 9.*

Proposition 3. *Given a (n,k) Reed–Solomon code, which has minimum distance $d = n-k+1$, one can guarantee an increase in error correction capability at each level of the tree provided that $k_{ij} - k_{i(j+1)} \geq 2, \forall\, j = 0, \ldots, T-1$.*

Proof. We must prove that $t_{i(j+1)} \geq t_{ij} + 1, \ \forall \ j = 0, \ldots, T-1$. For simplicity but without loss of generality, set $j = 0$. If $k_{i0} - k_{i1} \geq 2$, then we can write:

$$\begin{aligned} (-d_{i0} + n + 1) + d_{i1} - n - 1 &\geq 2 \\ d_{i1} - 1 &\geq d_{i0} - 1 + 2 \\ \left[\frac{d_{i1}-1}{2}\right] &\geq \left[\frac{d_{i0}-1}{2}\right] + 1 \\ t_{i1} &\geq t_{i0} + 1. \end{aligned}$$

This completes the proof. □

The verification that, for cyclic codes, there will not always be an increase in the error correction capacity between the levels of the tree, as considered in [12], leads us to search for answers on how to properly choose the generating polynomials for a code of parameters (n,k) and its subcodes, in order to guarantee subcodes with larger Hamming distance, with the purpose of observing an increase in the error correction capacity between the levels of the tree. An approach to constructing chains of some linear block codes while keeping the minimum distances (of the generated subcodes) as large as possible is presented in [25] and may be the solution to this problem.

3.3. An Example with a BCH Code

According to Luo and Vinck [25], to construct a chain of BCH subcodes with the characteristic of maintaining the minimum distance as large as possible, the task becomes more difficult because their subcodes may not be BCH and cyclic codes, and therefore the minimum distance of these subcodes might not be found easily. However, for primitive BCH codes, the minimum distance coincides with the weight of the generator polynomial, which makes it feasible to use it for the construction of the nested subcode chain that we seek. For non-primitive BCH codes, this statement is not always valid. For an extensive description of the minimum distance for BCH codes, we recommend consulting [26].

In Table 1, we present the parameters for binary primitive BCH codes of length $n = 2^m - 1$; it will guide the tree construction.

Table 1. Parameters for values of $m \leq 6$.

	$m=3$	$m=4$			$m=5$					$m=6$										
n	7	15			31					63										
k	4	11	7	5	26	21	16	11	6	57	51	45	39	36	30	24	18	16	10	7
t	1	1	2	3	1	2	3	5	7	1	2	3	4	5	6	7	10	11	13	15

Note that there will always be an increase in error correction capability for a fixed n and varying k.

Example 10. *Consider the root node associated with the BCH code $\mathcal{C}_{i0}(15,11)$. Suppose that we want to encode the packets $p_1(x) = x^3 + x^2 + x$, $p_2(x) = x$ and $p_3(x) = x^4 + x^2 + 1$ associated with nodes whose dimensions are $k_{t1} = 4$, $k_{t2} = 2$ and $k_{i2} = 5$, respectively. The polynomials $g_{i0}(x) = x^4 + x + 1$, $g_{i1}(x) = x^8 + x^7 + x^6 + x^4 + 1)$ and $g_{i2}(x) = x^{10} + x^8 + x^5 + x^4 + x^2 + x + 1$ generate the codes associated with the internal nodes, $\mathcal{C}_{i0}(15,11)$, $\mathcal{C}_{i1}(15,7)$ e $\mathcal{C}_{i2}(15,5)$, respectively, as shown in Figure 11.*

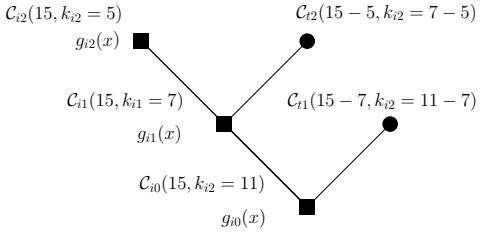


Figure 11. Tree construction of a BCH code tree.

Encoding the packets, we have:

$$\begin{aligned} C_1(x) &= p_1(x)g_{i0}(x) \\ &= x^7 + x^6 + x^5 + x^4 + x; \\ C_2(x) &= p_2(x)g_{i1}(x) \\ &= x^9 + x^8 + x^7 + x^5 + x; \\ C_3(x) &= p_3(x)g_{i2}(x) \\ &= \alpha^5 x^6 + \alpha^5 x^5 + \alpha^5 x^4 + \alpha^5 x^3 + \alpha^5 x^2 + \alpha^5 x + \alpha^5. \end{aligned}$$

The transmitted codeword $C_0(x)$ is given by:

$$\begin{aligned} C_0(x) &= C_1(x) + C_2(x) + C_3(x) \\ &= x^{14} + x^8 + x^7 + x^6 + x^3 + x + 1 \end{aligned}$$

Alternatively, it is possible to represent the codeword in vector form:

$$C_0(x) = (1,0,0,0,0,0,1,1,1,0,0,1,0,1,1)$$

After the error correction process, which will be performed on the sum of the coded packets, taking into account the side information available at each receiver, data recovery will occur as follows:

- $p_1(x) = \dfrac{(x^{14} + x^8 + x^7 + x^6 + x^3 + x + 1) \mod (x^8 + x^7 + x^6 + x^4 + 1)}{(x^4 + x + 1)}$;
- $p_2(x) = \dfrac{(x^{14} + x^8 + x^7 + x^6 + x^3 + x + 1) \mod (x^{10} + x^8 + x^5 + x^4 + x^2 + x + 1)}{(x^8 + x^7 + x^6 + x^4 + 1)}$;
- $p_3(x) = \dfrac{(x^{14} + x^8 + x^7 + x^6 + x^3 + x + 1)}{(x^{10} + x^8 + x^5 + x^4 + x^2 + x + 1)}$.

Remark 9. *We have made available at [20] an m-file Matlab algorithm that performs the tree construction operations and the data recovery for the BCH code.*

4. Index Coding with Multiple Interpretations

In the problem of index coding with multiple interpretations, we assume that receivers demand all the messages from the source and that the sender is unaware of the subset of messages already known in the receivers—performing an encoding so that any side information may be used by the decoder, in order to increase its error correction capability. Otherwise, if a receiver has no side information, the decoder considers the received word to belong to the highest rate code associated with the root node of the tree.

The proposed encoding process is shown in Figure 12 and can be performed in four main steps:

1. Encoding of the different data packets with nested cyclic codes, which consists of subdividing the vector space of a linear block code into vector subspaces, using each of them for encoding a different user;
2. Implementation of index coding at the relay node; the basic idea is that the different data packets, encoded by polynomial multiplications with linearly independent generators, are added and then forwarded to the receivers;
3. Multiple interpretations at the receivers that occur at the error correction stage, where each receiver can decode the received message at different rates depending on the known side information;
4. The data recovery stage, i.e., the process of decoding $C_1(x), \ldots, C_T(x)$ through polynomial operations (1), as described in Section 3.1.

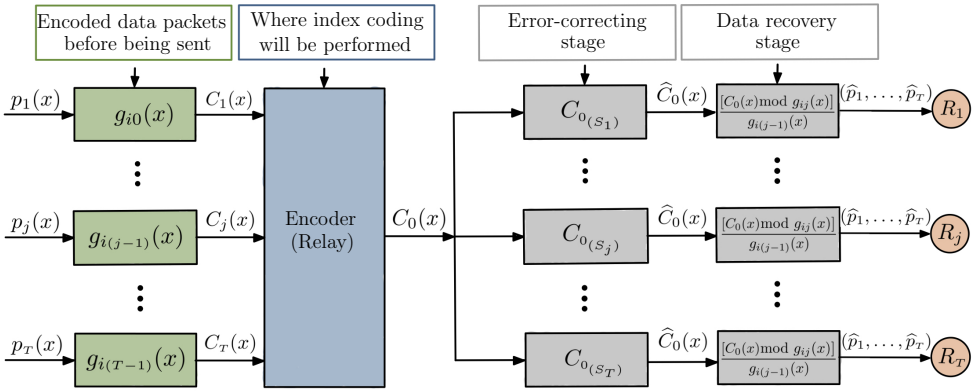


Figure 12. Scheme for index coding with multiple interpretations.

The notion of multiple interpretations was introduced in [13], indicating that the error correction capability in decoding gradually improves as the amount of side information available at the receiver increases. However, as we prove in Proposition 2, because of the nested structure of the tree, this characteristic of variable error correction capability can only be observed if there is a sequential removal of packets associated with the nodes, i.e., the side information should occur sequentially from the root to the top of the tree. However, in practice, this is not always the case. Thus, if we want to ensure that any information can be used efficiently in the decoder, it will be necessary to assume knowledge of the side information by the relay node or even the demand set, if we have a PICOD problem.

The following is a proposal for pliable index coding with multiple interpretations.

Pliable Index Coding with Multiple Interpretations

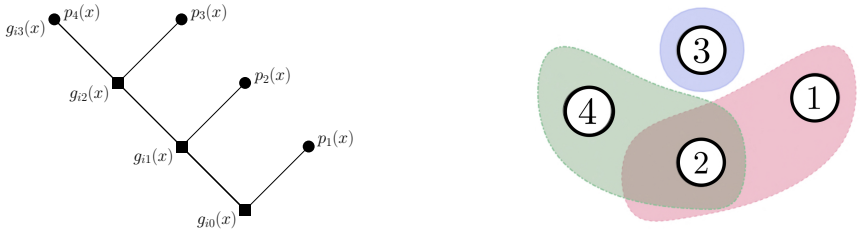
As in the pliable index coding problem [6], we will assume that the transmitter knows the demand set of each receiver and that all receivers are satisfied by receiving any message contained in their demand set. For example, if we are searching on the internet for a red flower image and we already have some previously downloaded pictures on our computer, if we find any other image that we do not have yet, we will be satisfied.

The goal of the server is to find an encoding scheme that satisfies all receivers, using as few transmissions as possible and ensuring that all side information associated with nodes

located below the node where the packet to be recovered is located may be interpreted as a gain in error correction capability, even when they do not appear in such a sequence.

The idea behind this proposal is to apply conflict-free coloring to the hypergraph that represents the demands of all receivers, and instead of sending the encoded word $C_0(x) = C_1(x) + C_2(x) + \cdots + C_T(x)$, we select the packets in a way that maximizes the possibility of a gain in error correction capability, since, as mentioned above, packages will only be removed if they occur sequentially.

Example 11. *Consider an instance of an pliable index coding with multiple interpretations in Figure 13a, where the encoded packets* $C_1(x) = g_{i0}(x)p_1(x), C_2(x) = g_{i1}(x)p_2(x), C_3(x) = g_{i2}(x)p_3(x)$ *and* $C_4(x) = g_{i3}(x)p_4(x)$ *will be sent to receivers* R_1, R_2 *and* R_3*, which have demand sets* $I_1 = \{1,2\}$, $I_2 = \{3\}$, $I_1 = \{2,4\}$*, respectively, as we see in Figure 13b.*



(**a**) (**b**)

Figure 13. Pliable index coding with multiple interpretations: (**a**) The construction representation of an instance of a pliable index coding problem with $m = 3$ (receivers) and $n = 4$ (messages). (**b**) Shows the hypergraph that describes this instance.

Figure 14 shows conflict-free coloring with two colors and $G_c = \begin{bmatrix} 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 \end{bmatrix}$, which represents the pliable index code.

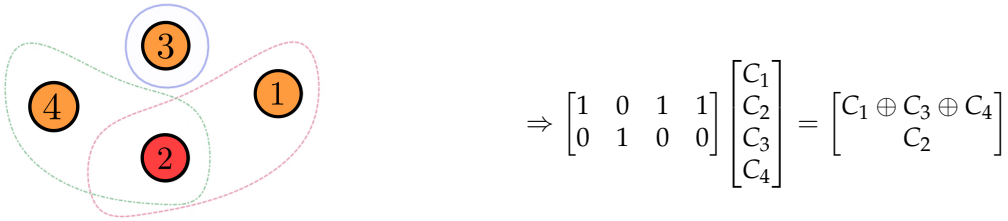


Figure 14. Conflict-free coloring with two colors.

Note that if we send only the message $C_0 = C_1 \oplus C_3 \oplus C_4$, all receivers recover one and only one message from their request set, as we can see in Table 2.

Table 2. Receivers with their side information sets.

Receivers	Side Information Sets	Decodes from Transmission
R_1	$S_1 = \{3,4\}$	$C_1 \oplus C_3 \oplus C_4 \oplus C_3 \oplus C_4 = C_1$
R_2	$S_2 = \{1,2,4\}$	$C_1 \oplus C_3 \oplus C_4 \oplus C_1 \oplus C_4 = C_3$
R_3	$S_3 = \{1,3\}$	$C_1 \oplus C_3 \oplus C_4 \oplus C_1 \oplus C_3 = C_4$

Each receiver R_j is satisfied if it can successfully recover any new message that is not present in its side information set, i.e., any message $x_d \in x_{I_j}$, where $I_j \triangleq [n] \setminus \mathcal{S}_j$.

Depending on the problem, this would be an ideal solution, since the transmitter may want each receiver to decode *only one message*, in which case we would have a PICOD(1); no client can receive more than one message from its request set. The case of PICOD(1) is

dealt with in detail in [27], and the following example, which aptly illustrates its use, is provided.

Consider a media service provider whom we pay for movies. The provider has a set of movies and customers pay for a certain number of movies, e.g., one movie. Suppose that the service is being sold in such a way that customers will be happy to receive any movie that they have not watched yet. There is a restriction on the service provider's side, since customers who have paid for only one movie should not receive more than one. Therefore, it can only supply one film for each client.

5. Conclusions

The verification that, for cyclic codes, there will not always be an increase in the error correction capacity between the levels of the tree leads us to search for ways to correctly choose the generating polynomials for a code and its subcodes, in order to guarantee subcodes with larger Hamming distance and an increase in error correction capability in consecutive levels of the tree. A method for the construction of chains of some linear block codes that maintains the minimum distances (of the generated subcodes) as large as possible is presented in Vinck and may be useful in addressing this issue.

Our work deals with the construction of index coding. We treat index coding as a network coding problem and we show how it is possible to construct pliable index codes with multiple interpretations by exploiting the conflict-free coloring of a hypergraph. Studying conflict-free coloring of a hypergraph in the context of the general index coding problem seems to be an interesting direction for future studies.

Author Contributions: Conceptualization, V.G.P. and M.H.M.C.; Formal analysis, V.G.P. and M.H.M.C.; Investigation, V.G.P.and M.H.M.C.; Methodology, V.G.P. and M.H.M.C. All authors have read and agreed to the published version of the manuscript.

Funding: This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior—Brasil (CAPES)—Finance Code 001.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Birk, Y.; Kol, T. Informed-source coding-on-demand (ISCOD) over broadcast channels. In Proceedings of the IEEE INFOCOM'98, the Conference on Computer Communications, Seventeenth Annual Joint Conference of the IEEE Computer and Communications Societies, San Francisco, CA, USA, 29 March–2 April 1998; Volume 3, pp. 1257–1264. [CrossRef]
2. Bar-Yossef, Z.; Birk, Y.; Jayram, T.S.; Kol, T. Index Coding With Side Information. *IEEE Trans. Inf. Theory* **2011**, *57*, 1479–1494. [CrossRef]
3. Unal, S.; Wagner, A.B. A Rate–Distortion Approach to Index Coding. *IEEE Trans. Inf. Theory* **2016**, *62*, 6359–6378. [CrossRef]
4. El Rouayheb, S.; Sprintson, A.; Georghiades, C. On the index coding problem and its relation to network coding and matroid theory. *IEEE Trans. Inf. Theory* **2010**, *56*, 3187–3195. [CrossRef]
5. Effros, M.; El Rouayheb, S.; Langberg, M. An Equivalence Between Network Coding and Index Coding. *IEEE Trans. Inf. Theory* **2015**, *61*, 2478–2487. [CrossRef]
6. Brahma, S.; Fragouli, C. Pliable Index Coding. *IEEE Trans. Inf. Theory* **2015**, *61*, 6192–6203. [CrossRef]
7. Kurose, J.F.; Ross, K.W. *Computer Networking: A Top-Down Approach*, 5th ed.; Addison-Wesley Publishing Company: San Francisco, CA, USA, 2009.
8. Song, L.; Fragouli, C. Making Recommendations Bandwidth Aware. *IEEE Trans. Inf. Theory* **2018**, *64*, 7031–7050. [CrossRef]
9. Song, L.; Srinivasavaradhan, S.R.; Fragouli, C. The benefit of being flexible in distributed computation. In Proceedings of the 2017 IEEE Information Theory Workshop (ITW), Kaohsiung, Taiwan, 6–10 November 2017; pp. 289–293. [CrossRef]
10. Song, L.; Fragouli, C.; Zhao, T. A Pliable Index Coding Approach to Data Shuffling. *IEEE Trans. Inf. Theory* **2020**, *66*, 1333–1353. [CrossRef]
11. Dau, S.H.; Skachek, V.; Chee, Y.M. Error Correction for Index Coding With Side Information. *IEEE Trans. Inf. Theory* **2013**, *59*, 1517–1531. [CrossRef]
12. Barbosa, F.C.; Costa, M.H.M. A tree construction method of nested cyclic codes. In Proceedings of the 2011 IEEE Information Theory Workshop, Paraty, Brazil, 16–20 October 2011; pp. 302–305. [CrossRef]
13. Xiao, L.; Fuja, T.E.; Kliewer, J.; Costello, D.J. Nested codes with multiple interpretations. In Proceedings of the 2006 40th Annual Conference on Information Sciences and Systems, Princeton, NJ, USA, 22–24 March 2006; pp. 851–856. [CrossRef]

14. Alencar, V.P.; Costa, M.H.M. Index coding from Reed-Solomon Codes. In Proceedings of the Brazilian Society of Computational and Applied Mathematics, SBMAC, São Paulo, Brazil, 10–11 November 2021; Volume 8. [CrossRef]
15. Blahut, R.E. *Algebraic Codes for Data Transmission*, 1nd ed; Cambridge University Press: Cambridge, UK, 2003. [CrossRef]
16. Pless, V. FJ MacWilliams and NJA Sloane, The theory of error-correcting codes. I and II. *Bull. Am. Math. Soc.* **1978**, *84*, 1356–1359. [CrossRef]
17. Alon, N.; Hasidim, A.; Lubetzky, E.; Stav, U.; Weinstein, A. Broadcasting with side information. In Proceedings of the 2008 49th Annual IEEE Symposium on Foundations of Computer Science, Philadelphia, PA, USA, 25–28 October 2008; pp. 823–832. [CrossRef]
18. Lubetzky, E.; Stav, U. Nonlinear Index Coding Outperforming the Linear Optimum. *IEEE Trans. Inf. Theory* **2009**, *55*, 3544–3551. [CrossRef]
19. Peeters, R. Orthogonal representations over finite fields and the chromatic number of graphs. *Combinatorica* **1996**, *16*, 417–431. [CrossRef]
20. Alencar, V.G.P. Construcao de Arvore em MatLab. 2021. Available online: https://github.com/valeriaurca/Construcao-de-Arvore.git (accessed on 4 June 2022).
21. Krishnan, P.; Mathew, R.; Kalyanasundaram, S. Pliable Index Coding via Conflict-Free Colorings of Hypergraphs. In Proceedings of the 2021 IEEE International Symposium on Information Theory (ISIT), Melbourne, Australia, 12–20 July 2021; pp. 214–219. [CrossRef]
22. Even, G.; Lotker, Z.; Ron, D.; Smorodinsky, S. Conflict-Free Colorings of Simple Geometric Regions with Applications to Frequency Assignment in Cellular Networks. *SIAM J. Comput.* **2003**, *33*, 94–136. [CrossRef]
23. Heegard, C. Partitioned linear block codes for computer memory with 'stuck-at' defects. *IEEE Trans. Inf. Theory* **1983**, *29*, 831–842. [CrossRef]
24. Marvasti, F.; Hasan, M.; Echhart, M.; Talebi, S. Efficient algorithms for burst error recovery using FFT and other transform kernels. *IEEE Trans. Signal Process.* **1999**, *47*, 1065–1075. [CrossRef]
25. Han Vinck, A.J.; Luo, Y. Optimum distance profiles of linear block codes. In Proceedings of the 2008 IEEE International Symposium on Information Theory, Toronto, ON, Canada, 6–11 July 2008; pp. 1958–1962. [CrossRef]
26. Lin, S.; Costello, D.J. *Error Control Coding: Fundamentals and Applications*, 2nd ed.; Pearson-Prentice Hall: Upper Saddle River, NJ, USA, 2004.
27. Sasi, S.; Rajan, B.S. On pliable index coding. *arXiv* **2019**, arXiv:1901.05809.