Technical University of Crete

Department of Electronic and Computer Engineering

Zero-error Varying-length Distributed Source Coding

Diploma Thesis

by

Nikolaos Kolomvakis

Advisor

Assistant Professor George N. Karystinos

July 2010

Contents

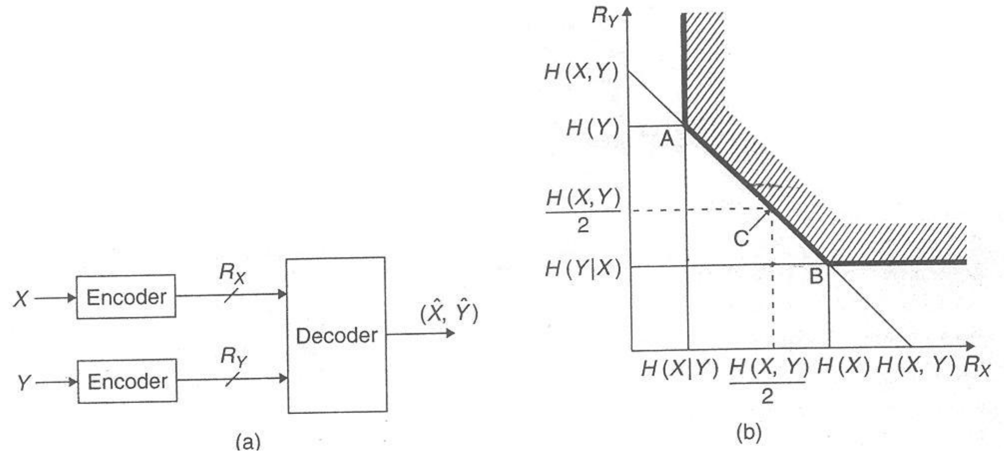

Figure 1: Distributed source coding of statistically dependent i.i.d discrete random sequences X and Y. (a) Setup; (b) Achievable rate region

I. Introduction

One of the enabling technologies for sensor networks is distributed source coding (DSC), which refers to the compression of multiple correlated sensor outputs that do not communicate with each other (hence distributed coding). These sensors send their compressed outputs to a central point (e.g., the base station) for joint decoding[12]. Here, we focus on two sensors S_X and S_Y which send corellated data to a third sensor or a central processing unit S_Z, without communicating with each other. More specifically, we study the region of achievable rates R_X and R_Y, if the reconstruction at S_Z is to be lossless. Generally, the sensors could be operating under three different requirements[10], which are listed below:

1. *Coding with vanishingly small error*: This is the original DSC scheme as introduced by Slepian and Wolf and its rate region is given by $R_X \geq H(X|Y), R_Y \geq H(Y|X)$ and $R_X + R_Y \geq H(X, Y)$ (Fig. 1(b)). It requires only that $Pr\{(\hat{X}^n, \hat{Y}^n) \neq (X^n, Y^n)\} \to 0$ as $n \to \infty$

2. *Varying-length zero-error* (VLZE) *coding*: This scheme requires that $Pr\{(\hat{X}^n, \hat{Y}^n) \neq (X^n, Y^n)\} = 0$, $\forall n \geq 1$. As a result, the achievable rate region is smaller.

3. *Fixed-length zero-error coding*: These zero-error codes are required to be fixed-length. Unlike the simple case in point-to-point coding communication, where the achievable rate is simply the logarithm of the size of source alphabet, if the correlation is sufficiently strong between the sequences that sensors convey to the third sensor, then it can still be exploited for compression.

Here, we focus on the asymmetric coding, a special case where the decoder S_Z knows Y and the goal is to uniquely describe X using the smallest possible average rate and discuss its connection to graph theory, which provides the most suitable tool for the problem. For this case, the problem is described by the characteristic graph and the marginal probability mass function P_X. The problem reduces to designing codes such that no vertex can be assigned the same codeword with its neighbors. We discuss VLZE coding in detail. We will see two classes of VLZE codes (Unrestricted Inputs and Restricted Inputs) and what is the performance relation between them through graph entropies. Finally, we discuss coding techniques for both classes and we implement an optimal VLZE code design algorithm proposed in[4] and a fast suboptimal design algorithm for Unrestricted Inputs codes.

II. Basic Definitions and Notations

A graph $G = (X, E)$ comprises a set X of vertices together with a set E of edges, which are 2-element subsets of X. For instance, the pentagon graph has a set of vertices $X = \{0, 1, 2, 3, 4\}$ and a set of edges $E = \{(0, 1), (1, 2), (2, 3), (3, 4), (4, 0)\}$.

Let $P_2(A)$ denote the set of all edges in a set of vertices A. In the pentagon graph example, we have $P_2(X) = E \cup \{(0, 2), (0, 3), (1, 3), (1, 4), (2, 4)\}$.

We say that $G(A)$ is a subgraph of $G = (X, E)$ with $A \subset X$ if its vertex set is A and its edge set is $E \cap P_2(A)$. For example the graph G_S with vertices $\{0, 1, 2\}$ and edges $\{(0, 1), (1, 2)\}$ is subgraph of the pentagon graph.

For the graphs $G_1(X, E_1)$ and $G_2(X, E_2)$, we use the notation $G_1 \subset G_2$ to indicate that $E_1 \subset E_2$. In the same way, we denote by $G_{G_1 \cup G_2} = G_1 \cup G_2$ the graph with vertex set X and edge set $E_1 \cup E_2$.

The complement graph of $G = (X, E)$ is a graph $\bar{G}$ on the same vertices X such that two vertices of $\bar{G}$ are adjacent if and only if they are not adjacent in G. That is, $\bar{G} = (X, E^c)$, where $E^c = P_2(X) \setminus E$ (The notation "$P_2(X) \setminus E$" expresses the set difference between $P_2(X)$ and E and is defined as $P_2(X) \setminus E = \{x : x \in P_2(X)$ and $x \notin E)$. For example, the complementary of the pentagon graph is a graph $\bar{G} = (X, E^c)$ with a set of vertices $X = \{0, 1, 2, 3, 4\}$ and a set of edges $E^c = \{(0, 2), (0, 3), (1, 3), (1, 4), (2, 4)\}$.

An *empty* graph, denoted by E_X is a graph with no edges. A *complete* graph, denoted by K_X, is the graph in which each pair of vertices is connected by an edge. An example of a complete graph is $G_K = (X, P_2(X))$. It is obvious that $\bar{E_X} = K_X$ and $\bar{G} \cup G = K_X$.

An *independent* set X' is a set of vertices in a graph G if $G(X') = E'_X$. In the pentagon graph, an independent set is $X' = \{0, 2\}$ where $G(X') = E_X$. The set of all independent sets in G is denoted by $\Gamma(G)$. For instance, we present the set of all independent sets in the pentagon graph. Hence $\Gamma(G) = \{(0), (1), (2), (3), (4), (0, 2), (0, 3), (1, 3), (1, 4)\ ,(2, 4)\}$.

The cardinality of the largest independent set in G is called *stability number*, and it is denoted by $\alpha(G)$. According to the set $\Gamma(G)$, for the pentagon graph, we have $\alpha(G) = 2$.

In the same way, a set X' is called a *clique* in G if $G(X') = K_{X'}$ and the set of all cliques is denoted as $\Omega(G)$. An example of a complete graph is $G_K = (X, P_2(X))$. In this case, $\Omega(G_K)$ consists of all sets of vertices X', where $X' \subseteq X$ in G_K. For the pentagon graph, $\Omega(G) = \{(0), (1), (2), (3), (4), (0, 1), (0, 4), (1, 2), (2, 3), (3, 4)\}$. It is obvious that $\Omega(G) = \Gamma(\bar{G})$.

A *probabilistic graph* (G, P) consists of a graph and a random variable distributed over its vertices. As an example, let consider the pentagon graph with uniform distribution over its vertices. The probability (or weight) of each vertex equals $\frac{1}{5}$.

The AND-product of $G_1(X_1, E_1)$ and $G_2(X_1, E_2)$, denoted by $G_1 \times G_2$, has vertex set the cartesian product of the vertex sets X_1 and X_2, $X_1 \times X_2$ and its edge set results from the following rule: two distinct vertices (x_1, x_2) and (x'_1, x'_2) are connected if either $x_i = x'_i$ or x_i is adjacent to x'_i in graph G_i for each $i = 1, 2$.

The OR-product of G_1 and G_2, denoted by $G_1 \cdot G_2$, has the vertex set $X_1 \times X_2$, but distinct vertices (x_1, x_2) and (x'_1, x'_2) are connected if either x_1 is adjacent to x'_1 or x_2 is adjacent to x'_2 in graphs G_1 and G_2, respectively.

We denote by $G^n = (X^n, E_n)$ and $G^{(n)} = (X^n, E_{(n)})$, the nth AND- and OR- products (or nth AND- and OR-powers) of G respectively. Note that $\bar{G^{(n)}} = \bar{G}^n$. Moreover, $G^n \subset G^{(n)}$.

Finaly, the strongly typical set of sequences $T^n_{P_x,\epsilon}$ is defined as

$$T^n_{P_x,\epsilon} = \{x^n \in X^n : |\frac{1}{n}N(\alpha|x^n) - P_X(\alpha)| \leq \epsilon\}$$

where $N(\alpha|x^n)$ is the number of occurances of α in x^n. We use the fact that strong typicality captures most of the probability:

$$P^n_X(T^n_{P_x,\epsilon}) \geq 1 - \frac{|X|}{4n\epsilon^2}. \tag{1}$$

Let X, Y be discrete random variables with alphabets $\mathscr{X}$ and $\mathscr{Y}$ respectively and let $\{(X_i, Y_i)\}_{i=1}^{\infty}$ be a sequence of independent drawings of a pair of dependent random variables X,Y. Here a pair (X,Y) distributed over a countable product set $\mathscr{X} \times \mathscr{Y}$ according to a probability $p(x, y)$. We desire to encode the sequence $\{X_i\}$ such that the decoder can decode it without error. We assume that the decoder has access to the side infotmation $\{Y_i\}$.

Formally, the *support set* (S) of (X,Y) is the set:

$$S \triangleq \{(x, y) : p(x, y) > 0\}$$

of possible (x, y) pairs. Distinct $x, x' \in \mathscr{X}$ are *confusable*, written $x \eqsim x'$, if there is a $y \in \mathscr{Y}$ such that $(x, y),\ (x', y) \in S$.

$x\backslash y$	0	1	2	3	4
0	0.1	0.1	0	0	0
1	0	0.1	0.1	0	0
2	0	0	0.1	0.1	0
3	0	0	0	0.1	0.1
4	0.1	0	0	0	0.1

Figure 2: The joint distribution P_{XY} of the example 1

Example 1: We Consider $\mathscr{X} = \mathscr{Y} = \{0, 1, 2, 3, 4\}$ with $P_X(x) = \frac{1}{5}\ \forall\ x \in \mathscr{X}$ and

$$P_{Y|X}(y|x) = \begin{cases} \frac{1}{2} & y = x \ or \ y = x + 1 \ mod\ 5 \\ 0 & \text{otherwise} \end{cases}$$

We construct the joint probability mass function (p.m.f) $p(x, y)$ according to the formula: $p(x, y) = p(y|x)p(x)$ as it is shown in Fig. 2. The support set for this joint p.m.f is $S = \{(0, 0), (0, 1), (1, 1), (1, 2), (2, 2), (2, 3), (3, 3), (3, 4), (4, 0), (4, 4)\}$. We see that given the side-information y, the value of x is not determined. For instance, when $y = 0$ then x can be either 0 or 1. In addition, given $y = 4$, we see that x can be either 0 or 4. We say that the letter 0 is confusable with the letters 1 and 4.

This confusability relation of the pair (X, Y) can be captured in the *characteristic graph* G. A characteristic graph $G = (\mathscr{X}, E)$ has vertices the alphabet set $\mathscr{X}$ and $\{x, x'\} \in E$ if distinct $x, x' \in \mathscr{X}$ are confusable. The characteristic graph of the example 1 is shown in Fig. 3.

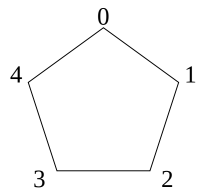

Figure 3: The characteristic graph of the example 1

III. Coding

A. Fixed-length zero-error coding

In any valid code, two confusable letters may not be assigned the same codeword. In the example 1, a valid encoder must assign different codewords between the symbol 0 and the symbols 1, 4. However, it is possible to be assigned the same codeword to symbols 1 and 4, since they are not confusable.

As an example of an encoding scheme, we color the vertices of the characteristic graph such that no two adjacent edges share the same color (this is known as vertex coloring). In other words, we choose a set Z, whose each element $z_i \in \Gamma(G)$, such that $\bigcup_{i=1}^{K} z_i = \mathscr{X}$ and $\bigcap_{i=1}^{K} z_i = \emptyset$ with $1 \leq K \leq |\Gamma(G)|$. Then, we assign a fixed-length codeword to each color (FLZE coding). In Fig. 5(a), is shown a FLZE coding for the characteristic graph of the the example 1. We do not care about the encoding of the side information Y, since it is known to the decoder. Note that given side information y, decoder is able to identify uniquely the vertex x, among all vertices with the same color, that satisfies $P_{XY}(x, y) > 0$, since the symbols that occupy the same color are not confusable.

For instance, consider the support set of example 1. A possible set Z that satisfy the vertex coloring is: $\{(0, 2), (1, 3), (4)\}$ (It is illustrated in Fig. 5(a)). According to the joint p.m.f. if side information $y = 1$,x can be either 1 or 2. Therefore, it is possible for the encoder to communicate either the independent set (color) $c_1 = \{0, 2\}$ or $c_2 = \{1, 3\}$. In both cases, decoder is able to decode uniquely the symbol either it is 1 or 2. Notice that if we assigned the same color in both symbols 1 and 2, decoder would not be able to decide which one has been sent, since both are possible.

The joint decoder for this scheme is shown in Fig. 4. In a FLZE scheme, the minimum achievable coding rate is given by $\lceil \log_2 \chi(G) \rceil$, where $\chi(G)$ denotes the *chromatic number* of graph G. The *chromatic number* of a graph G is defined as the minimum number of colors needed for a valid coloring of G. In the example we

$\phi_X(x)\backslash\phi_Y(y)$	c_1	c_2	c_3	c_4	c_5
c_1	(0,0)	(0,1)	(2,2)	(2,3)	$(*, *)$
c_2	$(*, *)$	(1,1)	(1,2)	(3,3)	(3,4)
c_3	(4,0),	$(*, *)$	$(*, *)$	$(*, *)$	(4,4)

Figure 4: Joint decoder for the optimal FLZE scheme of the example 1. The symbol $*$ stands fo a "don't care" output

discussed previously, we achieve the optimal vertex coloring. Notice that the chromatic number of a graph yield the minimum coding rate only in FLZE scheme. We will see that it is not necessary on the VLZE codes.

In Fig. 5(a) is shown this optimal FLZE coloring scheme that uses $\chi(G) = 3$ colors (we can assign the codewords 00, 01, and 11 to the colors c_1, c_2, c_3 respectively). Hence, the achieving rate is $\lceil \log_2 \chi(G) \rceil = 2$ bits per symbol.

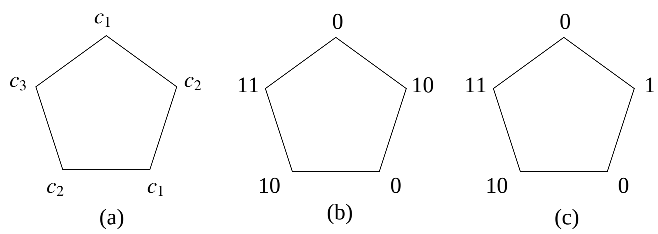

Figure 5: The characteristic graph in (a) shows the optimal coloring scheme, and (b) shows the resultant UI code obtained by encoding the colors in (a). By RI code, a better rate can be obtained as is shown in (c)

B. Varying-length zero-error coding

Varying-length codes for the side-information problem were introduced by Alon and Orlitsky[1]. They defined two families of binary variable-length codes:

1 A *restricted inputs* (*RI*) code for (G, P) is a mapping $\phi : X \rightarrow \{0, 1\}^*$ such that if $\{x, x'\} \in E$ *then* $\phi(x)$ is not a prefix of $\phi(x')$.

2 An *unrestricted inputs* (*UI*) code for (G, P) is a mapping $\phi : X \rightarrow \{0, 1\}^*$ such that, for every distinct pair $x, x' \in V$, $\phi(x)$ is not a proper prefix of $\phi(x')$ and if $\{x, x'\} \in E$ then $\phi(x) \neq \phi(x')$.

It is obvious that UI codes is subclass of the class of RI codes. It has been shown that an RI code may be expressed as a coloring of the characteristic graph, followed by one-to-one encoding of the colors. Similarly, a UI code is a coloring of G followed by a prefix-free coding of the colors. However, Note that in neither of these cases does the optimal code necessarly induce a coloring with the minimum number of colors. For example, consider the 3-colorable graph in Fig. 6(a). The optimal binary RI code, which, in this case, is the same as the optimal UI code, induces a coloring with four colors in Fig. 6(b)[4].

The expected number of bits transmitted under the distribution $p(x)$ by $\phi(x)$ is:

$$\bar{\ell}(\phi) \triangleq \sum_{x \in X} P(x)|\phi(x)| \tag{2}$$

where $|\phi(x)|$ is the length of the string $\phi(x)$.

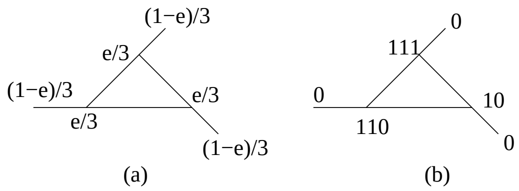

Figure 6: The node labels in (a) indicate probabilities with $e \leq 1/4$. In (b) they indicate optimal codewords

We denote by $\bar{L}(G, P)$ and $\bar{\mathscr{L}}(G, P)$ the minimum rate of an RI and UI code for (G,P) respectively.

$$\bar{L}(G, P) = min\{\bar{\ell}(\phi) : \phi\ is\ an\ RI\ code\ for\ (G, P)\}$$

$$\bar{\mathscr{L}}(G, P) = min\{\bar{\ell}(\phi) : \phi\ is\ an\ UI\ code\ for\ (G, P)\}$$

and the codes attaining these minima are called optimal codes.

General, we have that

$$\bar{L}(G, P) \leq \bar{\mathscr{L}}(G, P) \tag{3}$$

Example 2: For the characteristic graph of the example 1, we assign a UI code:

$$Pr\{X = 0\} = Pr\{X = 2\} = \frac{1}{5}, \qquad \phi(0) = \phi(2) = 0$$
$$Pr\{X = 1\} = Pr\{X = 3\} = \frac{1}{5}, \qquad \phi(1) = \phi(3) = 10$$
$$Pr\{X = 4\} = \frac{1}{5}, \qquad \phi(4) = 11$$

The expected length in this scheme is $\bar{\ell} = \frac{1}{5}\{|\phi(0)| + |\phi(1)| + |\phi(2)| + |\phi(3)| + |\phi(4)|\}$
$= \frac{1}{5}\{1 + 2 + 1 + 2 + 2\} = 1.6$ bits

However, we can achieve a better rate performance if we assign in the same pentagon graph a RI code: $\phi(0) = \phi(2) = 0, \phi(1) = 1, \phi(3) = 10$ and $\phi(4) = 11$. In this case the expected length is $\bar{\ell} = \frac{1}{5}\{|\phi(0)| + |\phi(1)| + |\phi(2)| + |\phi(3)| + |\phi(4)|\} = \frac{1}{5}\{1 + 1 + 1 + 2 + 2\} = 1.4$ bits

We can see that an optimal RI code can achieve better rate performance that an optimal UI code.

Example 3[1]: For $\epsilon \in [0,1)$ let (X,Y) be distributed over $\{1,...,n\} \times \{1,...,n\}$ according to

$$P_\epsilon(x,y) = \begin{cases} \frac{1-\epsilon}{n} & , y = x \\ \frac{\epsilon}{n^2-n} & , x \neq y \end{cases}$$

When $\epsilon = 0$, then X = Y, this imply that we do not need to comunicate letters from the source to the join decoder, since we know them a priori via side information y, hence $\bar{L} = 0$. In this case the characteristic graph is empty. When $\epsilon > 0$ any two distinct elements of $\{1,...,n\}$ are confusable, hence $\bar{L} \geq log\ n$, with equality when n is a power of 2. Here, the characteristic graph is complete.

Note that for the distribution p_ϵ the charactiristic graph G constists of the vertex set $\{$1,...,n$\}$ and the random variable X is distributed uniformly over its vertices (This can be shown computing the marginal probability mass function $p_X(x) = \sum_{y \in \{1,...,n\}} P_\epsilon(x,y) = \frac{1}{n}$).

When Y is independent of X (e.g. when Y is constant or inexistent) classical results show that

$$H(X) \leq \bar{L} \leq H(X) + 1 \tag{4}$$

where

$$H(X) \triangleq \sum_{x \in \mathscr{X}} p(x) log_2 \frac{1}{p(x)}$$

is the binary *entropy* of the chance variable X. For instance, let X be a discrete random variable with alphabet $\mathscr{X} = \{1,2,3,4\}$ and probability mass function $Pr\{X = x\} = \frac{1}{4}, \forall x \in \mathscr{X}$. Then $H(X)$ = 2 bits.

For general (X,Y) , the only known bounds are

$$H(X|Y) \leq \bar{L} \leq H(X) + 1 \tag{5}$$

where

$$H(X|Y) \triangleq \sum_{y \in \mathscr{Y}} p(x) \sum_{x \in \mathscr{X}} p(x|y) log \frac{1}{p(x|y)}$$

is the *conditional binary entropy* of X given Y.

C. Block coding

The encoder for a block code divides the information sequence into message blocks, each message block contains n independent instances (information symbols) over an alphabet set $\mathscr{X}$. In this case, a message could be represented as n-tuple $x^n = (x_1, x_2, ..., x_n) \in \mathscr{X}^n$ and the total number of possible different messages is $|\mathscr{X}|^n$. The encoder transforms a message x^n onto a codeword:

$$\phi_n : \mathscr{X}^n \to \{0, 1\}^*$$

We extend the notion of confusability to vectors. Hence, distinct

$$x^n = (x_1, x_2, ..., x_n) \in \mathscr{X}^n$$
$$x'^n = (x'_1, x'_2, ..., x'_n) \in \mathscr{X}^n$$

are confusable if and only if every distinct pair $(x_i, x'_i), i = 1, 2, ..., n$ is confusable. The characteristic graph for (X^n, Y^n) is, as we define above, the nth AND-products of the graph G for (X,Y) and it is denoted as $G^n = (\mathscr{X}^n, E_n)$, for the underlying probability mass function:

$$P^n_{XY}(x^n, y^n) = \prod_{t=1}^{n} P_{XY}(x, y).$$

We have already studied fixed-length zero-error coding with blocklength $n = 1$. Now, consider $n = 2$, that is $x^2 = (x_1, x_2)$. We provide the characteristic graph G^2 in Fig. 7, where the five individual pentagons represent the confusability relation for x_2, each for fixed x_1, and the meta-pentagon (see each individual pentagon as a node x_1 connected with the other two adjacent pentagons) represents the confusability relation for x_1 alone. There are a total of 100 edges in G^2. For the sake of simplicity, we only present the edges that show the conusability of the letter $(3, 0)$.

According to the joint p.m.f for $n = 1$ in Fig. 2, we see that the symbol $x^2 = (3, 0)$ can be produced for y^2 = (3,0), (3,1), (4,0) and (4,1). Thus, the support set (x^2, y^2) is: for $y^2 = (3, 0)$ we have $x^2 = \{(2, 0), (2, 4), (3, 0), (3, 4)\}$, for $y = (3, 1)$ we have x^2 =(2,0),(2,1),(3,0) and (3,1), for y^2=(4,0) we have x^2 =(3,0),(3,4),(4,0) and (4,4) and for y^2 =(4,1) we have x^2 =(3,0),(3,1),(4,0) and (4,1). We see that (3,0) is confusable with the following letters: (2,0), (2,1), (2,4),(3,1), (3,4), (4,0), (4,1) and (4,4). This implies that G^2 is in fact the same as the nth AND-power of G. It can be shown that[10]:

$$c(x_1, x_2) = 2x_1 + x_2 \; mod \; 5$$

is a valid code coloring scheme. This shows the benefit of fixed-length zero-error block-coding even since the bitrate per symbol drops to $\frac{1}{2}\lceil log_2\chi(G)\rceil = 1.5$ bits.

D. Varying-length zero-error block coding

The previous definitions of RI and UI codes for (G, P) may now be extended to RI and UI block codes for (G^n, P^n), where P^n is the product distribution induced on $\mathscr{X}^n$ by P

$$P^n((x_1, x_2, ..., x_n)) = \prod_{i=1}^{n} P(x_i)$$

Let $\bar{L}_n(\bar{\mathscr{L}}_n)$ is the total expected number of bits that encoder P_X must transmit in the RI (UI) scenario. We

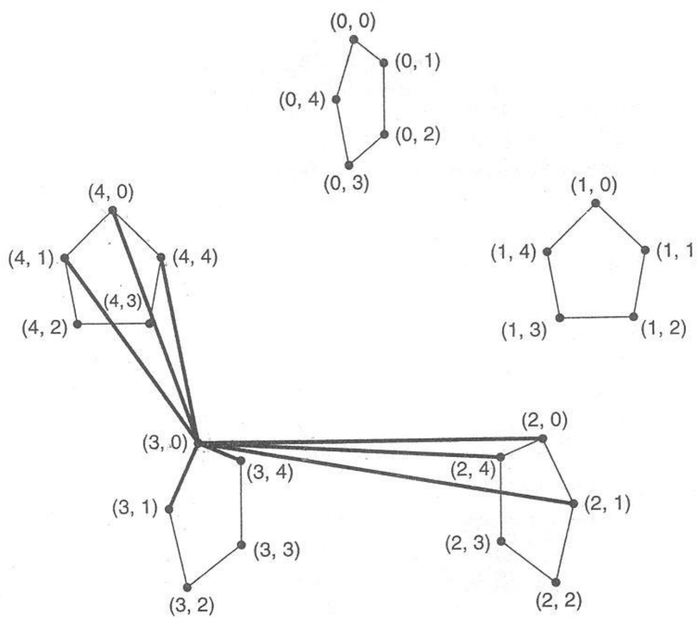

Figure 7: Partial illustration of the graph G^2

have for every (X,Y) pair and every n that[1]:

$$\bar{L}_n \leq \bar{\mathscr{L}}_n \tag{6}$$

The bit rate expended by a RI code block code is determined by

$$R_{RI}^n = \frac{1}{n} \sum_{x \in X^n} P_X^n(x^n) |\phi_n(x^n)| \tag{7}$$

Once characteristic graph G is bult and P_X is known, there is no further dependence of the minimum rate on $P_{Y|X}$. Therefore, the minimum achievable rate with blocklength n will henceforth be denoted as $\bar{R}_{RI}^n(G, P_X)$.

We are interested in the number of bits required for a large number of instances. Let define the following quantities

$$\bar{R}_{RI} \triangleq \lim_{n\to\infty} \frac{1}{n} \bar{L}_n$$

$$\bar{R}_{UI} \triangleq \lim_{n\to\infty} \frac{1}{n} \bar{\mathscr{L}}_n$$

which expresses the expected per-instance number of bits that must be transmitted for an asymptotically large number of instances of RI and UI codes respectively.

We determine $\bar{R}_{UI}$ and show that $\bar{R}_{RI}$ can be remarkably smaller than either $\bar{L}$ or $\bar{R}_{UI}$.

IV. Properties of VLZE codes

A. Graph Entropies

Definition 4.1. The graph enrtropy of G under P_X is defined as:

$$H(G, P_x) = \lim_{n\to\propto} \frac{1}{n} log_2[\min_{A:P_x^n(A)>1-\epsilon} \chi(G^{(n)}(A))] \tag{8}$$

In words, $H(G, P_X)$ is the normalized logarithm of the minimum number of colors needed to color any high probability subset of $G^{(n)}$. Exactly, how high the probability, which is determined by the value of $0 < \epsilon < 1$, is irrelevant[10]. $K\ddot{o}$rner also derived a single-letter characterization of $H(G, P_X)$, given by:

$$H(G, P_x) = \min_{\substack{overall \\ pairs\ (x,z)}} \{I(X; Z) : X \sim P_x, X \in Z \in \Gamma(G)\} \tag{9}$$

where

$$I(X; Z) \triangleq H(Z) - H(Z|X)$$

is the *mutual information* between X and Z and $X \in S \in \Gamma(G)$ is a notation for $z \in \Gamma(G)$ and $P_{Z|X(z|x)} = 0, \ if x \notin z$.

Let X and Z be random variables distributed over a countable product set $\mathscr{X} \times \mathscr{Z}$ according to a joint probability distribution $p(x, z)$. X defines a probability distribution over the vertices of G. For every vertex x we select a transition probability distribution $p(x|z)$ ranging over the independet sets that contain $x : p(z|x) \geq 0$ and

$$\sum_{z \ni x} p(z|x) = 1$$

This specifies a joint distribution of X and a random variable Z ranging over the independent sets and always containing X. The graph entropy of G is the smallest possible mutual information between X and Z. It is obvious that $0 \leq H(G, P_x) \leq H(X)$ for all (G,X), thus $0 \leq I(X; Z) \leq H(X)$ for all (X,Z).

Example[1]: For the empty graph, the set of all vertices is independent and always contain X, thus the graph entropy is 0. For the complete graph, the only independent set is that containing exactly one vertex, thus we must have $Z = \{X\}$ giving $H(G, P_X) = I(X; Z) = H(X)$. In the pentagon graph, every independent set contains one or two vertices, hence $H(X|Z) \leq 1$ implying that $I(X, Z) \geq H(X) - 1$. If X is distributed uniformly over the vertices, we can let $p(z|x) = \frac{1}{2}$ for each of the two-element independent sets containing vertex x. Then, by symmetry, $H(X|Z) = 1$, implying that $H(G, P_X) = log_2 5 - 1 \approx 1.32$.

Definition 4.2: The complementary graph enrtropy of G under P_X is defined as:

$$\bar{H}(G, P_x) = \lim_{\epsilon \to 0} \bar{H}_\epsilon(G, P_x)$$

Where

$$\bar{H}_\epsilon(G, P_X) = \lim_{n \to \propto} Sup \frac{1}{n} log_2[\chi(G^n(T^n_{P_x, \epsilon}))] \tag{10}$$

The complementary graph entropy imply that G^n has a high-probability induced subgraph which can be colored with approximately $2^{n\bar{H}(G,P)}$ colors. It is very similar to that of graph entropy, except that AND-powers instead of OR-powers are colored. This difference prohibited a single-letter formula for $\bar{H}(G, P_x)$. It even remains unknown whether $\lim_{\epsilon \to 0}$ is necessary or whether *limsup* can be replaced by a regular limit.

Definition 4.3: The chromatic entropy of a probabilistic graph (G,X) is given by:

$$H_\chi(G, P_X) = min\{H(c(x)) : \text{c}(\cdot) \text{ is a valid coloring of G}\} \tag{11}$$

the lowest entropy of any coloring of G.

Example[1]: The empty graph can be colored with one color, thus has chromatic entropy 0. The complete graph requires a different color for every vertex hence has chromatic entropy $H(X)$. The pentagon graph with uniform distribution over the vertices requires three colors, one assigned to a single vertex and each of the other two assigned to two vertices, thus the chromatic entropy $H(0.4, 0.4, 0.2) \approx 1.52$. The pentagon graph with distribution $p_0 = 0.3, p_1 = p_2 = p_4 = 0.2$ and $p_3 = 0.1$ achieves its lowest coloring entropy when the color classes are $\{0, 2\}, \{1, 2\}$ and $\{3\}$

Definition 4.4: The *clique* entropy of a probabilistic graph (G,X) is given by:

$$H_\omega(G, P_x) = \max_{x \in z \in \Omega} H(X|Z) = H(P_x) - H(\bar{G}, P_x) \tag{12}$$

which is intimately related to the graph entropy.

Example[1]: For the empty graph, the only cliques are singletons, thus $Z = \{X\}$, therefore we have that $H_\omega(G, P_x) = 0$. For the complete graph, we can consider Z as the set of all vertices. In this case $H_\omega(G, P_x) = H(X)$. Now, we consider the pentagon graph with uniform distribution over the vertices. Every clique contains one or two vertices. Hence $H_\omega(G, P_x) \leq 1$. On the othe hand, if for every x we let Z be uniformly distributed over the two-element clique sets containing x, then by symmetry $H(X|Z) = 1$, implying that $H_\omega(G, P_x) = 1$.

Definition 4.5: The capacity of the probabilistic graph (G, P_X) is given by:

$$C_\epsilon(G, P_X) = \lim_{n \to 0} sup \frac{1}{n} log_2[\alpha(G^n(T^n_{P_{x,\epsilon}}))] \tag{13}$$

and

$$C(G, P_X) = \lim_{\epsilon \to 0} C_\epsilon(G, P_X) \tag{14}$$

$$\tag{15}$$

We will need the following relation between $\bar{H}(G, P)$ and $C(G, P)$ for proving theorem 4.7:

$$\bar{H}(G, P) + C(G, P) = H(P) \tag{16}$$

B. VLZE code bounds

Next, we provide some known lower and upper bounds for RI and UI codes.

Theorem 4.1: For every source (X, Y) pair:

$$H_\chi(G, X) \leq \bar{\mathscr{L}} \leq H_\chi(G, X) + 1 \tag{17}$$

Theorem 4.2: For RI codes, messages must be prefix-free only over graph edges. It has been proved that for every source (X, Y) pair:

$$H_\chi(G, X - log_2[H(X) + 1] - log_2 e \leq \bar{L} \leq H_\chi(G, X) + 1 \tag{18}$$

Theorem 4.3: For the single instance case, that is blocklength $n = 1$, it has been showed that for every source pair (X, Y):

$$\bar{L} \geq H(G, P) \tag{19}$$

Combining theorems 4.2 and 4.3, we see that for all source pairs:

$$H(G, P_X) \leq \bar{L} \leq H_\chi(G, P_X) \tag{20}$$

Theorem 4.4: Additionaly, it is known that for every source pair (X, Y) we have:

$$H(X|Y) \leq H_\omega(G, X) \leq H(G, X) \tag{21}$$

Theorem 4.5: For every probabilistic graph (G, P)

$$H(G, P) \leq H_\chi(G, P) \tag{22}$$

Proof[1]: We provide the proof of this theorem because it sheds some light on the the intuition behind graph theorem. Let Z be a random variable that ranges disjointly over $\Gamma(G)$ and we write $Z \mathring{\in} \Gamma(G)$, if Z attains disjoint values in $\Gamma(G)$. If c is a coloring of G then $Z \triangleq c^{-1}(c(X))$ express the color class of X. Conversely, every random variable that ranges disjointly over $\Gamma(G)$ and always contains X can be viewed as the color class of X. In this case, X determines Z, thus $H(Z|X) = 0$ and therefore

$$\begin{aligned} H_\chi(G, P_X) &= \min_{X \in Z \mathring{\in} \Gamma(G)} H(Z) \\ &= \min_{X \in Z \mathring{\in} \Gamma(G)} I(X; Z) \\ &\geq \min_{X \in Z \in \Gamma(G)} (X; Z) = H(G, P_X) \end{aligned}$$

Interpreting the proof, the chromatic entropy of a probabilistic graph is the minimum, over all color classes, of the information a vertex give us about its color. The graph entropy has the same interpretation, except that every vertex is now assigned a random color.

We can generalizate theorems 4.1 and 4.3 into VLZE block coding:

$$\frac{H_\chi(G^n, P^n)}{n} \leq \bar{R}^n_{RI} \leq \frac{H_\chi(G^n, P^n) + 1}{n} \tag{23}$$

$$\bar{R}^n_{RI} \geq \frac{H(G^n, P^n)}{n} \tag{24}$$

Additionaly, Alon and Orlitsky[1] showed the following relations (theorem 4.6 and 4.7) between the asymptotic minimum rate for RI and UI codes and graph entropy:

Theorem 4.6:

$$\bar{R}_{RI} \leq H(G, P) \tag{25}$$

Theorem 4.7:

$$\bar{R}_{UI} = H(G, P) \tag{26}$$

Theorem 4.8

$$\bar{R}_{RI}(G, P_x) = \lim_{n\to\propto} \frac{1}{n} H_\chi(G^n, P^n_X) \tag{27}$$

Proof[10]. The limit in (4.6) exists because $H_\chi(G^n, P^n_X)$ is subadditive in n[8]. The concatenation of asymptotically optimal zero-error variable-length codes for (G_1, P_{X_1}) and (G_2, P_{X_2}) yield a valid code for $(G_1 \times G_2, P_{x_1} P_{x_2})$ with rate $\bar{H}(G_1, P_{X_1}) + \bar{H}(G_2, P_{X_2})$. Since the asymptotically optimal code for $(G_1 \times G_2, P_{x_1} P_{x_2})$ can only perform better, hence:

$$H_\chi(G_1 \times G_2, P_{x1} \times P_{x2}) \leq H_\chi(G_1, P_{x1}) + H_\chi(G_2, P_{x2})$$

Therefore

$$H_\chi(G^n, P_x^n) \leq nH_\chi(G, P_x) \tag{28}$$

Alon and Orlitsky proved that

$$H_\chi(G^n, P^n) - log\{H_\chi(G^n, P^n) + 1\} - loge \leq nR_{RI}^n \leq H_\chi(G^n, P^n) + 1$$

Using the inequality (27) we have

$$H_\chi(G^n, P_x^n) - log_2[nH_\chi(G, P_x) + 1] - log_2 e \leq n\bar{R}_{RI}^n \leq H_\chi(G^n, P^n) + 1 \Longrightarrow$$

$$H_\chi(G^n, P_x^n) - log_2[nH_\chi(G, P_x) + n] - log_2 e \leq n\bar{R}_{RI}^n \leq H_\chi(G^n, P^n) + 1 \Longrightarrow$$

$$H_\chi(G^n, P_x^n) - log_2[H_\chi(G, P_x) + 1] - log_2 n - log_2 e \leq n\bar{R}_{RI}^n \leq H_\chi(G^n, P^n) + 1$$

normalized by n and taking limits we have the theorem 4.1.

Theorem 4.9.

$$\bar{R}_{RI}(G, P_x) = \bar{H}(G, P_x) \tag{29}$$

Proof[10]. we first show that:

$$\bar{H}(G, P_x) \geq \lim_{n\to\propto} \frac{1}{n} H_x(G^n, P_x^n) \tag{30}$$

Toward that end, we fix $\epsilon > 0$ and observe from equation (9) that $\forall n > n_0(\epsilon)\ \exists\ c(\cdot)$ so that

$$|c(T^n_{P_x,\epsilon})| \leqslant 2^{n(\bar{H}_\epsilon(G,P_x)+\epsilon)}$$

Define the idicator function $\Phi : X^n \rightarrow \{0, 1\}$:

$$\Phi(x^n) = \begin{cases} 1 & \text{if} x^n \in T^n_{P_{x,\epsilon}} \\ 0 & \text{otherwise} \end{cases}$$

We then have:

$$\begin{aligned} H_x(G^n, P^n_x) &\leqslant H(c(x^n)) \\ &\leqslant H(c(x^n)) + H(\Phi|c(x^n)) = H(\Phi) + H(c(x^n)|\Phi) \\ &= H(\Phi) + p(X \in T^n_{P_{x,\epsilon}})H(c(x^n)|x^n \in T^n_{P_{x,\epsilon}}) + p(X \notin T^n_{P_{x,\epsilon}}H(c(x^n)|x^n \notin T^n_{P_{x,\epsilon}}) \\ &\leqslant H(\Phi) + H(c(x^n)|x^n \in T^n_{P_{x,\epsilon}}) + \epsilon H(c(x^n)|x^n \notin T^n_{P_{x,\epsilon}}) \\ &\leqslant 1 + n[\bar{H}_\epsilon(G, P_x) + \epsilon(1 + log|X|)] \end{aligned}$$

where we used that $|c(T^n_{P_{x,\epsilon}})| \leqslant 2^{n(\bar{H}_\epsilon(G,P_x)+\epsilon)}$ in the last step. Normalized by n and taking limits, we have:

$$\begin{aligned} \lim_{n\to\propto} \frac{1}{n} H_\chi(G^n, P^n_x) &\leqslant \lim_{n\to\propto} [\frac{1}{n} + \bar{H}_\epsilon(G, P_x) + \epsilon(1 + log|X|)] \\ &= \bar{H}_\epsilon(G, P_x) + \epsilon(1 + log|X|) \\ &\overset{\epsilon\to 0}{=} \bar{H}(G, P_x) \end{aligned}$$

Now consider the reserved inequality in (29) . Fix $\epsilon > 0$ and let the coloring function c on G^n achieve $H_\chi(G^n, P^n_x)$, so that

$$H_\chi(G^n, P^n_x) = H(c(X^n))$$

To lower bound $H(c(X^n))$, we use the following elementary lower bound for the entropy function. If Q is a probability distribution over the set W and S $\subseteq$ W then:

$$H(Q) \geq -[\sum_{j \in S} Q(j)] log \max_{j \in S} Q(j) \tag{31}$$

$$\implies H(c(x^n)) \geq -P_x^n(T_{P_{x,\epsilon}}^n) log[\max_{x^n \in T_{P_{x,\epsilon}}^n} P_x^n(C(x^n))] \tag{32}$$

The probability $P_X^n(T_{P_{x,\epsilon}}^n)$ can be lower bounded as in (1). In any coloring of G^n, the maximum cardinality of a single-colored subset of $T_{P_{x,\epsilon}}^n$ cannot exceed $\alpha(G^n(T_{P_{x,\epsilon}}^n)$. Thus:

$$\max_{x^n \in T_{P_{x,\epsilon}}^n} P_x^n(C(x^n)) \leqslant \alpha(G^n(T_{P_{x,\epsilon}}^n) \max_{x^n \in T_{P_{x,\epsilon}}^n} P_x^n(x^n) \tag{33}$$

From[11] we have:

$$-\frac{1}{n} log \max_{x^n \in T_{P_{x,\epsilon}}^n} P_x^n(x^n) \geq [H(P_x) + \epsilon |X| log \epsilon] \tag{34}$$

substituting (1),(33) and (34) in 32:

$$\implies \frac{1}{n} H_x G^n, P_x^n \geq (1 - \frac{|X|}{4n\epsilon^2}\{H(P_x) - \frac{1}{n} log \alpha((G^n(T_{P_{x,\epsilon}}^n)) + \epsilon |X| log \epsilon\}$$

$$\implies \lim_{n \to \propto} \frac{1}{n} H_x(G^n, P_x^n) \geq H(P_x) - C_\epsilon(G, P_x) + \epsilon |X| log \epsilon$$

$$\overset{\epsilon \to 0}{=} H(P_x) - C(G, P_x) = \bar{H}(G, P_x)$$

$$\implies \bar{R}_{RI}(G, P_x) = \lim_{n \to \propto} \frac{1}{n} H_x(G^n, P_x^n) = \bar{H}(G, P_x)$$

From theorems 4.4, 4.6, 4.7 and 4.9 we have:

$$H_{\omega}(G, P_x) \leq \bar{H}(G, P_x) \leq H(G, P_x) \leq H_{\chi}(G, P_x)$$

Despite the fact that neither theorem 4.8 nor theorem 4.9 provides a single-letter characterization for $\bar{R}_{RI}(G, P_x)$, the latter gives important insights into the problem:

- It is known that $H_{\omega}(G, P_x) \leq \bar{H}(G, P_x) \leq H(G, P_x) \leq H_{\chi}(G, P_x)$. Therefore the single-letter lower and upper bounds for complemantary graph $\bar{H}(G, P_x)$ can be tranlated to bounds for the minimum rate $\bar{R}_{RI}(G, P_x)$

- The theorem 4.2 reveals an asymptotically optimal variable-length coding scheme, where we can encode all the vertices in $T^n_{P_x,\epsilon}$ using roughly $n\bar{H}(G, P_x)$ bits while the rest vertices with roughly $nlog|X|$ bits

C. Chromatic Entropy Approximation

Next, we focus on computing of the chromatic entropy of a graph. We propose a heuristic algorithm for computing the chromatic entropy.

As we have defined, a proper coloring c of a graph assigns colors to vertices such that adjacent verices have distinict colors. If c is a function defined over $\mathscr{X}$, then $c(X)$ is a random variable with entropy:

$$H(c(X)) = -\sum_{\gamma \in c(\mathscr{X})} P(c^{-1}(\gamma)) log_2 P(c^{-1}(\gamma))$$

Here, the function $c^{-1}(\gamma)$ returns the vertices that belong to color γ and $P(c^{-1}(\gamma)) = \sum_{x\in\gamma} P(x)$.

The minimum entropy coloring problem can be defined also as[6]:

Instance: An undirected graph $G = (\mathscr{X}, E)$

Solution: A proper coloring $\phi : \mathscr{X} \to \mathbb{N}$ of G

Objective: Minimize the entropy $-\sum_i p_i log p_i$, where $p_i := |\phi^{-1}(i)/|\mathscr{X}|$

In other words, our goal is to minimize $H(c(X))$ choosing disjoint independent sets $s_i \in \Gamma(G)$ such that covering $\mathscr{X}$. This problem can be written as the following integer linear programming (ILP) problem:

$$minimize \sum_{z\in\Gamma(G)} C(z)s_z \quad \text{where } C(z) = -p_z \log_2 p_z \text{ and } p_z = \sum_{x\in z} p_X(x) \ \ \text{(minimize the total cost)}$$

$$subject\ to \sum_{z\in\Gamma(G):x\in z} s_z = 1, \forall x \in \mathscr{X} \quad \text{(the independent sets must cover } \mathscr{X} \text{ and be disjoint)}$$

$$s_z \in \{0,1\}, \forall\, z \in \Gamma(G) \quad \text{(every independent set is either in the set cover or not)}$$

The minimum entropy coloring problem is hard to solve and to approximate in general. It has been proved that finding a minimum entropy coloring of a weighted graph (in our case, probabilistic graph) is strongly NP-hard[6].

In order to approximate this problem, we recast it from ILP into a linear programming problem where it is solved in polynomial time. Namely we relax the constrain $p_z \in \{0,1\}, \forall\, z \in \Gamma(G)$ into $0 \leq p_z \leq 1, \forall\, z \in \Gamma(G)$.

Now, The problem is:

$$\begin{aligned}
&\mathit{minimize} \sum_{z\in\Gamma(G)} C(z)s_z \quad \text{(minimize the total cost)}\\
&\mathit{subject\ to} \sum_{z\in\Gamma(G):x\in z} s_z = 1, \forall x \in \mathscr{X} \quad \text{(the independent sets must cover } \mathscr{X} \text{ and be disjoint)}\\
&\qquad 0 \le s_z \le 1, \forall\, z \in \Gamma(G) \quad \text{(constraint relaxation)}
\end{aligned}$$

Here, the vector s_z represent the conditional probability $p(z|x)$ with $\sum_{z\ni x} p(z|x) = 1$. After the minimization of the cost function, the vector x_s contains the weights of each independent set in the minimal value of the cost function.

In order to approximate chromatic entropy, we choose each independent set with the following priority:

1. it must have the greatest value in p_z from the nonselected independent set and
2. its elements must be disjoint with these of the selected independent sets.

In this way, we approximate the value and the coloring, $c(\cdot)$, of the chromatic entropy for each characteristic graph.

Recall that the difference between the chromatic and graph entropy is the constraint on the selection of the independent sets as we discussed in the proof of theorem 4.5. Here, despite the fact that, it is remarkable that the solution of the LP problem is the same as the theoretic result of the graph entropy for the characteristic graphs we provide in this work. In our future work, we will check if this is coincidence or not.

V. VLZE coding techniques

A. Lossless Side-Information Source Codes

Let X and Y be memoryless sources with joint probability mass function (p.m.f) $p(x, y)$ on a finite alphabet $\mathscr{X} \times \mathscr{Y}$. A multiple access source code (MASC) is a source code designed for joint source (X,Y), where the encoder for each source operates without knowledge of the other source; the decoder jointly decodes the encoded bit streams from both sources.

A lossless instantaneous MASC for a joint source (X,Y) consists of two encoders $\phi_X : \mathscr{X} \to \{0,1\}^\star$ and $\phi_Y : \mathscr{Y} \to \{0,1\}^\star$ and a decoder $\phi^{-1} : \{0,1\}^\star \times \{0,1\}^\star \to \mathscr{X} \times \mathscr{Y}$. Here, $\phi_X(x)$ and $\phi_Y(y)$ is the binary descriptions of x and y respectively, and the probability of decoding error is:

$$P_e = Pr(\phi^{-1}(\phi_X, \phi_Y)) \neq (X, Y)).$$

We focus on lossless source coding, where $P_e \triangleq 0$.

When Y is perfectly known to the decoder, the problem reduces to the side-information source code (SISC), and the goal is to uniquely decode X using the smallest possible average rate. This scenario describes MASCs where ϕ_Y encodes Y using a traditional code for p.m.f $p(y)$ (e.g Huffman coding) so that ϕ_X can encode X assuming that the decoder knows Y. In this case: $\phi^{-1} : \{0,1\}^\star \times \mathscr{Y} \to \mathscr{X}$. If the decoder can correctly reconstruct x_1 by only reading the first $|\phi_X(x_1)|$ bits of $\phi_X(x_1)\phi_X(x_2)\phi_X(x_3)$..., then it is a lossless instantaneous SISC.

We present the SISC Prefix Property which is fundamental for optimal RI code design:

Lemma 1 (SISC Prefix Property): Code ϕ_X is a lossless instantaneous SISC for X given Y iff for each x, x', y with $p(x, y) > 0$ and $p(x', y) > 0$, $\{\phi_X(x), \phi_X(x')\}$ is prefix free[2].

We use trees to illustrate the prefix relationships between codewords: the description of x can be a proper prefix of the description of x' (written $\phi_X(x) < \phi_X(x')$) if and only if x is an ancestor of x' in the tree, and the description of x and x' can be identical ($\phi_X(x) = \phi_X(x')$) if and only if x and x' occupy the same node of the corresponding tree.

The resulting trees are similar to Huffman code trees in the sense that all nodes desceding from a common

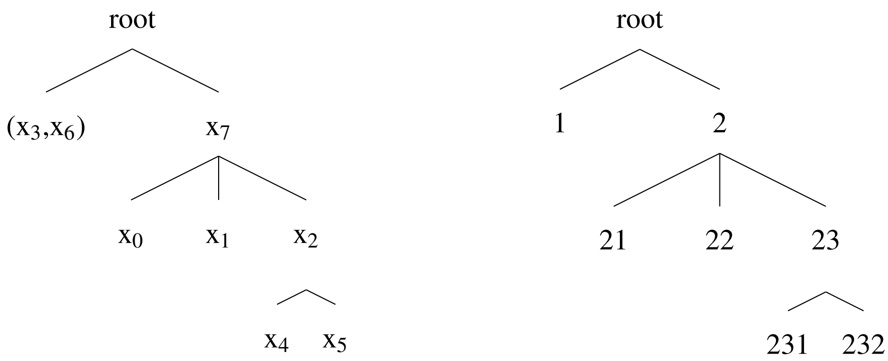

Figure 8: (a)Partition Tree $T(P(X))$. (b) Labels for $T(P(X))$

parent have descriptions that share a common prefix. However, they differ from Huffman trees in the point that they not need be binary or a symbol can be an internal node as well as leaves or multiple symbols can belong to the same node.

Groups, partitions and matched codes

Below we represent all necessary definitions[2] for constructing RI codes. These definitions rule out any construction that cannot yield a lossless instantaneous SISC.

An *one-level group* is a set $G = (x_1, ..., x_m)$ for any $p(x, y)$ if for any distinct $x_i, x_j \in G$, x_i is not confusable with x_j $(x_i \not\simeq x_j)$. The tree representation $T(G)$ for one-level group G is a single node representing all elements of G. For the probability mass function in Fig. 11, (x_1), (x_5, x_7) and (x_1, x_5, x_8) are all examples of valid *one-level groups*.

A *two-level group* for $p(x, y)$, comprises a root R and its children $C(R)$. R is one-level group, $C(R)$ is a set of one-level groups and, $G' \not\simeq R$ for all $G' \in C(R)$ where for any groups G_1 and G_2, G_1 is not confusable with G_2 if and only if $x_1 \not\simeq x_2$ for all $x_1 \in G_1$ and $x_2 \in G_2$. A two-level group is denoted by $G = (R : C(R))$. In the tree representation $T(G)$ for G, $T(R)$ is the root of $T(G)$ and the parent of all subtrees $T(G')$ for $G' \in C(R)$. An example of a two-level group for the probability mass function in table 1 is $G_2 = ((x_2) : \{(x_4), (x_5)\})$. The members of $C(R)$ are $\{(x_4), (x_5)\}$ and the members of G_2 are $\{(x_2), (x_4), (x_5)\}$. The tree representation $T(G_2)$ is a two-level tree consisting of a root and its two children, each of which is a single node.

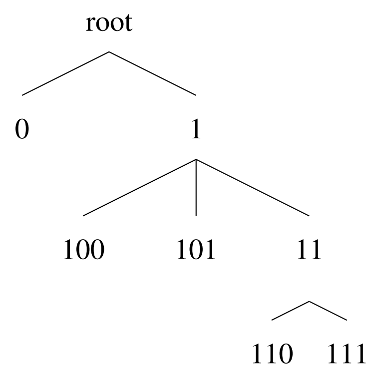

Figure 9: Matched code for $P(X)$

An *M-level group* for $p(x, y)$ for each $M > 2$ is a pair $G = (R : C(R))$ such that $G' \neq R$ for all $G' \in C(R)$. Here, R is a one-level group and $C(R)$ is a set of k-level groups with $k \leq M - 1$, at least one of them has (M-1) levels so that G can be M-level group. Again, $T(R)$ is the root of $T(G)$ and the parent of all subtrees $T(G')$ for $G' \in C(R)$. For any $M > 1$, an M-level group is also called multilevel group. An example of a *three-level* group for the same probability mass function as above is $G_3 = ((x_7) : \{(x_0), (x_1), ((x_2) : \{(x_4), (x_5)\})\})$. In $T(G_3)$, the root $T(x_7)$ of the three-level group has three children: the first two children are nodes (x_0) and (x_1) and the its last child is a two-level tree $T(x_2)$ with root the node (x_2) and children the nodes (x_4) and (x_5).

A *partition* $P(X)$ on a set X for p.m.f $p(x, y)$ is a set of nonempty subsets of X such that every every element in X is in exactly one of these subsets. That is, $P(X) = \{G_1, G_2, ..., G_m\}$ satisfies $\bigcup_{i=1}^{m} G_i = X$ and $G_j \bigcap G_i = \emptyset$ for any $j \neq k$, where each $G_i \in P(X)$ is a group for $p(x, y)$ and $G_j \bigcap G_i$ and $G_j \bigcup G_i$ refer to the intersection and the union respectively of the members of G_i and G_j. The tree representation of a partition is called a partition tree. The partition tree $T(P(X))$ for partition $P(X) = \{G_1, G_2, ..., G_m\}$ has an empty root $\mathbf{r}$ with m children, $T(G_1), ..., T(G_m)$. In Fig. 2(a), we provide a partition tree with partition: $P(X) = \{(x_3, x_6), G_3\}$

For any one-level group G at depth d in $T(P(X))$, let $\mathbf{n}$ describe the d-step path from root $\mathbf{r}$ to node $T(G)$ in $T(P(X))$ .We refer to G by describing this path. Thus, $T(\mathbf{n}) = T(G)$. For simplicity, we replace $\mathbf{n}$ for $T(\mathbf{n})$, when it is clear from the context that we are talking about the node rather than the one-level group at that node (e.g. we write $\mathbf{n} \in T(P(X))$ rather than $T(\mathbf{n}) \in T(P(X))$). To make the path descriptions unique in the whole tree, we fix an order on the descendants of each node and number them from left to right. Thus, the children of a node **n** are labeled as **n**1,**n**2,...,**n**K(**n**), where K(**n**) is the number of children descending from **n**. The labeled partition tree for Fig. 8(a) appears in Fig. 8(b).

The *node probability* $q(\mathbf{n})$ of one-level group **n** (or G) is the sum of the probabilities of that group's members. The subtree probability $Q(\mathbf{n})$ of one-level group **n** is the sum of probabilities the members of node **n** and descendants in $T(P(X))$. For the partition tre in Fig. 2(a), we have $q(23) = p_X(x_2)$ and $Q(23) = p_X(x_2) + p_X(x_4) + p_X(x_5)$.

A matched code ϕ_X for partition $P(X)$ is any binary code such that for any node $\mathbf{n} \in T(P(X))$ and symbols $x_1, x_2 \in \mathbf{n}$ and $x_3 \in \mathbf{n}k$ with $k \in \{1, ..., K(\mathbf{n})\}$ we have:

1. $\phi_X(x_1) = \phi_X(x_2)$
2. $\phi_X(x_1) < \phi_X(x_3)$
3. $\{\phi_X(\mathbf{n}k) : k \in \{1, ..., K(\mathbf{n})\}\}$ is prefix free.

An Example of a Huffman matched code for the partition appears in Fig. 9. Later, we will describe the methodology of Huffman coding construction in a partition tree with its corresponding probability mass function. For now, we focus on encoding and deconding of such a scheme. The encoder for the partition tree in Fig. 8(a) is:

$$\phi_X(x) = \begin{cases} 0 & \text{if } x \in \{x_3, x_6\} \\ 1 & \text{if } x \in \{x_7\} \\ 100 & \text{if } x \in \{x_0\} \\ 101 & \text{if } x \in \{x_1\} \\ 11 & \text{if } x \in \{x_2\} \\ 110 & \text{if } x \in \{x_4\} \\ 111 & \text{if } x \in \{x_5\} \end{cases}$$

The code achieves expected rate:

$$E_X|\phi_X(X)| = \sum_{\mathbf{n} \in T(P(X))} q(\mathbf{n}) l(\mathbf{n}) = 2,12 < H(X) = 2,91$$

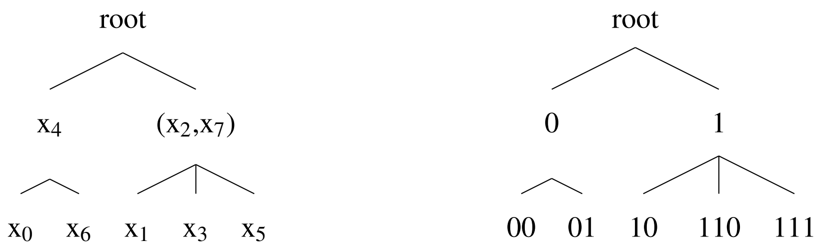

Figure 10: (a)optimal Partition Tree $T^*(P(X))$. (b) Codewords for $T^*(P(X))$

$x \backslash y$	y_0	y_1	y_2	y_3	y_4	y_5	y_6	y_7
x_0	.04	0	.04	.02	0	0	0	0
x_1	0	.04	0	0	.05	.1	0	0
x_2	.15	0	.05	0	0	0	0	0
x_3	0	.05	0	.06	0	0	0	0
x_4	0	.06	0	0	.05	0	0	0
x_5	0	0	0	.01	.02	.03	0	0
x_6	0	0	.01	0	0	.06	.02	.01
x_7	0	0	0	0	0	0	.05	.08

Figure 11: Sample p.m.f's on alphabet $\mathscr{X} \times \mathscr{Y}$ with $\mathscr{X} = \mathscr{Y} = \{x_1, ..., x_7\}$

by violating Kraft's inequality, since the code is non-prefix. Despite this fact, the code is uniquely decodable given side information Y. If we choose the partition more carefully, this rate can be reduced. For example, setting $\phi(0) = \phi(1) = 10$ in this code gives a lossles instantaneous code with expected rate 1,83. In next chapters, we investigate how we can select the optimal partition (Fig. 10(a)) tree and construct optimal RI codes (Fig. 10(b)).

Matched code design

we wish to design the optimal matched code for a partition $P(X)$ for $p(x, y)$. In traditional lossless coding, achieved by entropy encoding (e.g Huffman coding, Arithmetic coding), the optimal binary code length for a symbol is $l^{\star}(x) = -log_2 p(x)$ for all $x \in \mathscr{X}$ if those lengths are all integers. Below, we represent the theorem that gives the corresponding result for lossless SISCs on a partition $P(X)$.

Theorem 2: Given partition $P(X)$ for $p(x, y)$, the optimal matched code for $P(X)$ has code lengths $l^{\star}(\mathbf{r}) = 0$ and[2]

$$l^{\star}(\mathbf{n}k) = l^{\star}(\mathbf{n}) - log_2 \left(\frac{Q(\mathbf{n}k)}{\sum_{j=1}^{K(\mathbf{n})} Q(\mathbf{n}j)} \right)$$

for all $\mathbf{n} \in T(P(X))$ and $k \in \{1, ..., K(\mathbf{n})\}$ if those lengths are all integers.

Proof[2]: Given $x \in \mathbf{n} \in T(P(X))$, let $l(\mathbf{n}) = |\phi_X(x)|$. Then for any matched code ϕ_X for $P(X)$:

$$E\{\phi_X(X)\} = \sum_{\mathbf{n} \in T(P(X))} q(\mathbf{n}) l(\mathbf{n}) = \tag{35}$$

$$= \sum_{\mathbf{n} \in T(P(X))} \sum_{k=1}^{K(\mathbf{n})} Q(\mathbf{n}k)(l(\mathbf{n}k) - l(\mathbf{n})). \tag{36}$$

Thus, minimizing each term $\sum_{k=1}^{K(\mathbf{n})} Q(\mathbf{n}k)(l(\mathbf{n}k) - l(\mathbf{n}))$ independently, we can achieve the minimal expected rate. For sake of simplicity, we write $\phi_X(\mathbf{n}k) = cc_k$ for each $k \in \{1, ..., K(\mathbf{n})\}$, where $c = \phi_X(\mathbf{n})$ and c_k is the suffix of the kth descendant of $\mathbf{n}$. Thus, we have $|c_k| = l(\mathbf{n}k) - l(\mathbf{n})$. For $\{c_1, ..., c_{K(\mathbf{n})}\}$ to satisfy the prefix condition, in order to be uniquely decodable, $\{|c_1|, ..., |c_{K(\mathbf{n})}|\}$ must satisfy Kraft's inequality. Hence, the minimization of $\sum_{k=1}^{K(\mathbf{n})} Q(\mathbf{n}k)|c_k|$ subject to Kraft's inequality $\sum_{k=1}^{K(\mathbf{n})} 2^{-|c_k|} \leq 1$ is achieved by setting:

$$|c_k| = l(\mathbf{n}k) - l(\mathbf{n}) = -log_2 \left(\frac{Q(\mathbf{n}k)}{\sum_{j=1}^{K(\mathbf{n})} Q(\mathbf{n}j)} \right) \tag{37}$$

which completes the proof.

The proof of Theorem 2 demonstrates that we can design matched codes by designing entropy codes on the children of each internal node of a partition tree. All entropy coding algorithms are candidates for matched code design. We focus on matched Huffman. For any node $\mathbf{n}$ with $K(\mathbf{n}) > 0$, the huffman code $\phi^{(H)}_{X,P(X)}$ describes the step from $\mathbf{n}$ to $\mathbf{n}$k using a Huffman code designed for p.m.f.

$$\left(\frac{Q(\mathbf{n}k)}{\sum_{j=1}^{K(\mathbf{n})} Q(\mathbf{n}j)} \right)_{k=1}^{K(\mathbf{n})}$$

on alphabet $\{1, ..., K(\mathbf{n})\}$.

Example: We desire to build a matched Huffman code for the partition in Fig. 2, we work from top to bottom of the partition tree T. We start by designing a Huffman code for p.m.f:

$$\left(\frac{Q(k)}{\sum_{j=1}^{K(\mathbf{r})} Q(j)} \right)_{k=1}^{K(\mathbf{r})}$$

on the K($\mathbf{r}$ descendants of the root of tree T. We have that $K(\mathbf{r}) = 2$ $((x_3, x_6)$ and $(x_7))$ the p.m.f. is $(Q(1) + Q(2) = 1$, $p_X(x) = \sum_{y \in \mathscr{Y}} p(x, y), \forall x \in \mathscr{X})$:

$$\{p_X(x_3) + p_X(x_6), p_X(x_7) + p_X(x_0) + p_X(x_1) + p_X(x_2) + p_X(x_4) + p_X(x_5)\} = \{0.21, 0.76\}$$

The huffman code for the distribution $\{0.21,0.76\}$ is $\{0,1\}$. We repeat this process for each subsequent tree node $\mathbf{n}$ with $K(\mathbf{n}) > 0$. Node 2 has $K(\mathbf{n}) = 3$ and p.m.f $(Q_1 = \sum_{j=1}^{3} Q(2j) = 1 - Q(1) - p_X(7) = 1 - 0.21 - 0.13 = .066)$:

$$\{p_X(x_0)/Q_1, p_X(x_1)/Q_1, (p_X(x_2) + p_X(x_4) + p_X(x_5))/Q_1\} = \{0.1/Q_1, 0.19/Q_1, 0.37/Q_1\}$$

The Huffman code is $\{00, 01, 1\}$. Node 23 has $K(23) = 2$ and p.m.f ($Q_2 = p_X(x_4) + p_X(x_5) = 0.17$):

$$\{p_X(x_4)/Q_2, p_X(x_5)/Q_2\} = \{0.11/Q_2, 0.06/Q_2\}.$$

The huffman code is $\{0\ ,1\}$. At the end, $\phi_X(\mathbf{n})$ concatenates the Huffman codewords for all paths moving from $\mathbf{r}$ to $\mathbf{n}$ in T. The codewords for this example is shown in Fig. 2(c).

Theorem 3: Given a partition $P(X)$, matched Huffman codes for $P(X)$achieve the optimal expected rate over all matched codes for $P(X)$[2].

Proof: Let T be the partition tree of a partition $P(X)$. The codeword length of a node $\mathbf{n} \in T$ is denoted by $l(\mathbf{n})$ and the average length $\bar{l}$ for $P(X)$ is:

$$\bar{l} = \sum_{\mathbf{n}\in T} q(\mathbf{n})l(\mathbf{n}) = \sum_{k=1}^{K(\mathbf{r})}(Q(k)l(k) + \sum_{k\mathbf{n}\in T} q(\mathbf{n}k)(l(\mathbf{n}k) - l(k))$$

We denote:

$$\Delta l(\bar{k}) = \sum_{k\mathbf{n}\in T} q(\mathbf{n}k)(l(\mathbf{n}k) - l(k)).$$

for each $k \in \{1, ..., K(\mathbf{n})\}$.

Note that terms $\sum_{k=1}^{K(\mathbf{r})} Q(k)l(k)$ and $\Delta l(\bar{k})$ can be minimized independently. Thus,

$$min\bar{l} = min \sum_{k=1}^{K(\mathbf{r})} Q(k)l(k) + \sum_{k=1}^{K(\mathbf{r})} min\Delta l(\bar{k}).$$

In matched Huffman coding, working from the top to the bottom of the partition tree, we first minimize $\sum_{k=1}^{K(\mathbf{r})} Q(k)l(k)$ over all integer lengths $l(k)$ by using Huffman codes on $Q(k)$. We then minimize each $\Delta l(\bar{k})$

over all integer-length codes by traversing each level of tree T.

Optimal partitions: definitions and properties

Now, we focus on the partition yielding the best performance. Given a partition $P(X)$, let $l^{(H)}_{P(X)}$ and $l^{\star}_{P(X)}$ be the Huffman and optimal code lengths, respectively, for $P(X)$. We say that $P(X)$ is optimal for a matched Huffman SISC on $p(x, y)$ if

$$El^{(H)}_{P(X)} \le El^{(H)}_{P'(X)}$$

for any partition $P'(X)$ for $p(x, y)$ (The optimal partitions for matched matched and arithmetic SISCs can differ). Some properties of optimal partitions follow.

Lemma 2: There is an optimal partition $P^{\star}(X)$ for $p(x, y)$ for which every node except for the root of $P^{\star}(X)$ is nonempty and no node except for the root can have exactly one child[2].

Lemma 3: Let $T(\mathbf{n})$ be an arbitrary node in optimal partition $P^{\star}(X)$ for $p(x, y)$, and let $G = ((\mathbf{n}) : C(\mathbf{n}))$ be the group with root $\mathbf{n}$ and descendants identical to the descendants of $\mathbf{n}$ in $P^{\star}(X)$. Then, $\mathbf{n} = \{x \in G : \{x\} \not\asymp (G \cap \{x\}^c)\}$, in other words $\mathbf{n}$ is the maximum independent set of the group G $(\alpha(G))$. $C(\mathbf{n})$ is an optimal partition of $\{x \in G : x \notin \mathbf{n}\}$[2].

Partition design and complexity

We construct an optimal partition[2] for $\mathscr{X}$ by building optimal groups for larger and larger subsets $\mathscr{X}' \subseteq \mathscr{X}$ and testing all valid combinations of those groups. Let

$$R(X') = \{x \in G : \{x\} \neq (G \cap \{x\}^c)\}$$

be the root of a subtree for which each symbol $x \in X'$. We eliminate all X' with $R(X') = \emptyset$ by lemma 2. By lemma 3, the optimal group for X' is

$$G^\star(X') = (R(X') : C(X'))$$

where $C(X') = P^\star(X' \cap R(X')^c)$ is the optimal partition on $X' \cap R(X')^c$. The optimal group for a symbol is itself, $G^\star(\{x\}) = (x)$ for any $x \in X$. For any $X' \subseteq X$ with $|R(X')| > 0$ and $|X' \cap R(X')^c| > 0$, we find $C(X')$ by calculating the expected rate of the matched code for each set of groups of the form:

$$\begin{aligned} C = \Big\{ & G^\star(S_1), ..., G^\star(S_2) : |R(S_k)| > 0 \ \forall k, \\ & \bigcup_{k=1}^{K} S_k = (X' \cap R(X')^c), \\ & S_i \cap S_j = \emptyset, \ \forall i, j \Big\} \end{aligned}$$

and we choose the one with the best performance.

The number of the partitions for which we must design matched codes can be loosely bound from above by:

$$\sum_{k=1}^{|X|} \binom{|X|}{k} B_k < 2^{|X|} B_{|X|}$$

where $B_m \sim m^{-1/2}(\lambda(m))^{m+1/2} e^{\lambda(m)-m-1}$ is the number of ways a set of m elements can be partitioned into nonempty subsets and $\lambda(m)ln(\lambda(m)) = m$[13].

While the design is expensive, the encoding and decoding complexities for an optimal SISC are comparable to the encoding and decoding complexities of a traditional (single-sender, single-receiver) Huffman or arithmetic code. All are linear in $|X|$.[2]

B. Optimal design algorithm for RI codes

Previously, we designed an optimal coding algorithm for RI codes, using the language of partition trees. The problem is seperated into optimal code design for a given partition tree and search for the optimal partition tree. The necessary conditions for an optimal partition tree (Lemma 2 and Lemma 3), simplify the search for the optimal one.

Below, we provide an another approach[4] for searching the optimal RI codes using graphs. This technique is apparently simpler than the fisrt one.

For a probabilistic graph (G, P), let $G' = (X', E')$ be the subgraph induced in G by $X' \subseteq X$, we define:

$$P(X') = \sum_{x \in X'} P_X(x)$$

for the total probability of the set X' and

$$P_{X|X'}(x) = \frac{P_X(x)}{P(X')}$$

for the restricted distribution of P to X', $\forall x \in X'$.

We will use the following notation for the weighted codeword length of the subgraph $G(X')$:

$$L(X') = P(X')\bar{R}^1_{RI}(G(X'), P_{X|X'})$$

Let $\phi : X \to \{0,1\}^\star$ be the optimal RI code for (G, P). if i is a random intermediate node of the code tree that corresponds to ϕ, we define the following sets:

$$\phi^{-1}(i) = \{x \in X' : \phi(x) = i\}$$

$$\phi^{-1}(i*) = \{x \in X' : i \text{ is a prefix of } \phi(x)\}$$

Then for any code ϕ and codeword i, we have

$$L(\phi^{-1}(i*)) = P[\phi^{-1}(i0*)]\ \{\bar{R}^1_{RI}(G(\phi^{-1}(i0*)), P_{X|\{\phi^{-1}(i0*)\}}) + 1\} + P[\phi^{-1}(i1*)]\ \{\bar{R}^1_{RI}(G(\phi^{-1}(i1*)), P_{X|\{\phi^{-1}(i1*)\}}) + 1\}$$

which can be written as[4]:

$$L(\phi^{-1}(i*)) = L(\phi^{-1}(i0*)) + L(\phi^{-1}(i1*)) + P(\phi^{-1}(i*)) - P(\phi^{-1}(i))$$

where $i*$ provides that none of the sets are empty. Further, $i0*$ and $i1*$ denote the two children of i.

Since, ϕ is the optimal code, the last equation can be reformed as:

$$L(\phi^{-1}(i*)) = \min_{D \subseteq \phi^{-1}(i*) - \phi^{-1}(i)} \left\{ L(D) + L(\phi^{-1}(i*) - \phi^{-1}(i) - D) \right\} + P(\phi^{-1}(i*)) - P(\phi^{-1}(i))$$

The vertices in the set $\phi^{-1}(i)$ must be isolated in $G(\phi^{-1}(i))$, because otherwise they would insert ambiguity in the decoder. Thus, the equation above suggests an iterative algorithm to find R^1_{min}, and the corresponding optimal RI code. Let $I(X')$ be the set of isolated nodes in a induced subgraph of G, $G(X')$. Then, we have

$$L(X') = \min_{D \subseteq X' - I(X')} \left\{ L(D) + L(X' - I(X') - D) \right\} + P(X') - P(I(X')) \tag{38}$$

With the terminating condition that $L(X') = 0$, if $I(X') = X'$.

It is not necessary to search over all possible induced subsets $D \subseteq X' - I(X')$ in the minimization. It suffices to consider only those D that induce a dominating 2-partition, that is, where every vertex in D is connected to some vertex in $X' - I(X') - D$ and vice versa. This follows by the observations:

1. if a vertex in $\phi^{-1}(0*) - \phi^{-1}(0)$ is not connected to any vertex in $\phi^{-1}(1*)$, the rate can be reduced by assigning the codeword 1

2. A node in $\phi^{-1}(0)$ is not connected to any other node in $\phi^{-1}(0*)$, so if a vertex in $\phi^{-1}(0)$ is not connected to any vertex in $\phi^{-1}(1*)$, it is isolated in $G - I$, and the rate can again be reduced by moving it to I.

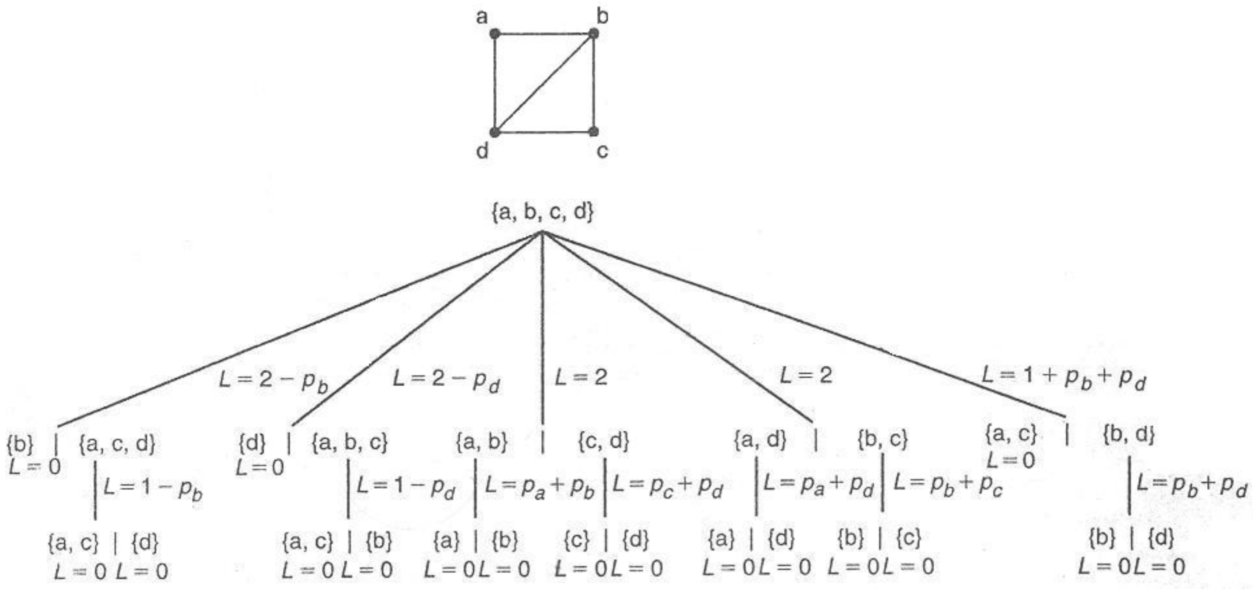

Figure 12: The recursive algorithm, which is terminated within two levels for this example. Values indicated on branches of the tree represent values of $L(X')$ returned for each 2-partition

Thus, the minimization can be restricted to the dominating 2-partitions of $G' - I$.

This recursive algorithm is illustrated in Fig. 12 using a small graph with vertices $\mathscr{X} = \{a, b, c, d\}$ and associated probabilities $\{p_a, p_b, p_c, p_d\}$. Initially, the algorithm splits the graph into a dominating 2-partition. Next, the algorithm splits each subgraph into a new dominating 2-partition until find a subgraph with isolated nodes.

For example, let $\{(b), (a, c, d)\}$ be a dominating 2-partition. In this case, we have:

$$L(\{a, b, c, d\}) = L(\{b\}) + L(\{a, c, d\}) + P(\{a, b, c, d\}) - P(I(\{a, b, c, d\}))$$

here, the subgraph $\{b\}$ is isolated (it is a single node),hence $L(\{b\}) = 0$ and $P(\{a, b, c, d\}) = 1$. There are not isolated nodes in $\{a, b, c, d\}$, therefore $P(I(\{a, b, c, d\})) = 0$. In the next step, the algorithm splits the subgraph $\{a, c, d\}$ into $(\{a, c\}, \{d\})$. Note, that this is the only dominating 2-partition among $\{a, c, d\}$.

Figure 13: (a) Optimal tree for RI code for the graph in Fig. 12. (b) codewords

Thus, we have:

$$L(\{a,b,c\}) = L(\{d\}) + L(\{a,c\}) + P(\{a,c,d\}) - P(I(\{a,c,d\}))$$

where $L(\{d\}) = L(\{a,c\}) = P(I(\{a,c,d\} = 0$ and $P(\{a,c,d\}) = 1 - p_b$. Therefore, we have $L(\{a,b,c,d\}) = 2 - p_b$.

In the same way, the algorithm splits the graph into all possible dominating 2-partition recursively as it is shown in Fig. 11. In the last step, it choose the path with the minimum rate $L(X)$.

As an example, we consider the p.m.f on Fig. 15(a). We have marginal probabilities $p_X(x)$: $p_a = p_b = p_c = p_d = 0,25$. In this case, the rate is minimized by choosing the path that split the initial graph into the dominating 2-partition ({a,c},{b,d}). We assign the codewords as they are shown below (Next, we will see analytically how we assign codewords in such a scheme). Thus, we have:

$$\phi_X(a) = \phi_X(c) = 0, \phi_X(b) = 10, \phi_X(d) = 11.$$

In this case, the rate is $L(X) = 1,5$ and it is coincide with $\mathscr{L}(\phi) = \bar{H}(G, P_X) = H(G, P_X)$.

On the other hand, if $p_a = \frac{1}{6}, p_b = \frac{1}{3}, p_c = \frac{1}{6}, p_d = \frac{1}{3}$ (joint p.m.f in Fig. 15(b)) the optimal code is achieved by spliting the initial graph into $(\{b\}, \{a,c,d\})$. In this case we have:

$$\phi_X(a) = \phi_X(c) = 10, \phi_X(b) = 0, \phi_X(d) = 11.$$

This optimal code achieves $L(X) = \frac{5}{3} \approx 1,66$, whereas $H(G, P_X) = 1,58$.

Figure 14: (a)Optimal path for UI code for the pentagon graph. (b) codewords

In Fig. 13(a) is shown the optimal tree, extracted from the recursive algorithm we discussed for RI codes if $p_a = p_b = p_c = p_d = 0.25$. The encoding is emlpoyed from root to leafs assigning the bit 0 at the left branch and the bit 1 at the rigth branch recursively for each level. Note, that the codeword of each symbol is updated concatenating its codeword level by level. The final codewords are produced when encoder traverse all nodes of the tree.

In Fig. 12 RI codes coincide with UI codes. On the other hand, in Fig. 14(a) we provide the tree for the optimal RI code derived from the joint p.m.f in Fig. 2. Its UI and RI coding is shown in Fig. 5(a) and Fig. 5(b) respectively.

Next, we outline a possible implementation of the optimal binary RI coding algorithm.

Input: (G, P)

1. for $r = 1 : |X|$
2. for $i = 1 : \binom{|X|}{r}$
3. I = find the isolated vertices in $G_{i,r}$
4. if set $\{G_{i,r} - I\}$ is empty go to step 9
5. for $k = 1 : \lfloor \frac{|G_{i,r} - I|}{2} \rfloor$
6. $D = \binom{G_{i,r} - I}{k}$
7. if $\{V - D, D\}$ is not dominating 2-partition go to step 9
8. Given k compute $L(V - D)$ and $L(D)$
9. end
10. Find the pair $(V - D, D)$ that minimize $L(G_{i,r})$ over all k
11. end
12. end

Note, that the the calculated optimal codes for smaller subgraphs may be used in the minimization of the step 8. The worst case complexity,C, of the algorithm is[4]:

$$C = \sum_{r=1}^{|V|} \binom{|V|}{r} O(2^r) = O(3^{|V|})$$

The complexity of $O(2^r)$ is a result of an exhaustive search over all possible smaller subgraphs of G.

$x\backslash y$	a	b	c	d
a	$\frac{1}{4}$	0	0	0
b	$\frac{1}{12}$	$\frac{1}{12}$	0	$\frac{1}{12}$
c	0	$\frac{1}{8}$	$\frac{1}{8}$	0
d	$\frac{1}{12}$	0	$\frac{1}{12}$	$\frac{1}{12}$

$x\backslash y$	a	b	c	d
a	$\frac{1}{6}$	0	0	0
b	$\frac{1}{9}$	$\frac{1}{9}$	0	$\frac{1}{9}$
c	0	$\frac{1}{12}$	$\frac{1}{12}$	0
d	$\frac{1}{9}$	0	$\frac{1}{9}$	$\frac{1}{9}$

Figure 15: Sample p.m.f's on alphabet $\mathscr{X} \times \mathscr{Y}$ with $\mathscr{X} = \mathscr{Y} = \{a, b, c, d\}$

Here, we provide an encoder for the optimal tree derived from the previous algorithm. :

encoding(**r**)

1. if **r**.left = **r**.right = null go to 4
2. left(**r**.left)
3. right(**r**.right)
4. return

left(**n**)

1. Concatenation of each symbol's bit stream in node **n** with bit 0
2. if **n**.left = **n**.right = null go to 5
3. left(**n**.left)
4. right(**n**.right)
5. return

right(**n**)

1. Concatenation of each symbol's bit stream in node **n** with bit 1
2. if **n**.left = **n**.right = null go to 5
3. left(**n**.left)
4. right(**n**.right)
5. return

C. Optimal design algorithm for UI codes

Based on the above recursive algorithm, we can construct optimal binary UI codes. In this case, the recursive relation, since the codeword set must be prefix free, is modified into[4]:

$$\mathscr{L}(X') = \min_{D \subseteq X'} \Big\{ L(D) + L(X' - D) \Big\} + P(X') \tag{39}$$

with terminating condition that $\mathscr{L}(X') = 0$, if $I(X') = X'$.

Next, we outline a possible implementation of the optimal binary UI coding algorithm.

Input: (G, P)

1. for $r = 1 : |X|$

2. for $i = 1 : \binom{|X|}{r}$

3. if set $\{G_{i,r}\}$ is empty go to step 10

4. for $k = 1 : \lfloor \frac{|G_{i,r}|}{2} \rfloor$

5. $D = \binom{G_{i,r} - I}{k}$

6. if $\{V - D, D\}$ is not dominating 2-partition go to step 9

7. Given k compute $L(V - D)$ and $L(D)$

8. end

9. Find the pair $(V - D, D)$ that minimize $L(G_{i,r})$ over all k

10. end

11.end

Here, as we see in Fig.13(a), the codewords occupy only the leafs of the optimal tree.On the contrary, codewords in a RI scheme can occupy also internal nodes.

The algorithm derived from these recursion has again worst case complexity $O(3^{|V|})$

$Rate \backslash n$	1	2
R^n_{UI}	1.60	1.20

$Rate \backslash n$	1	2	3
R^n_{UI}	1.66	1.61	1.60

Figure 16: (a) block coding performance for the joint p.m.f in Fig.2 (b) in Fig.15(b)

$p(x,y) \backslash R_X$	$H(X)$	$H(G,P)$	$H_{\chi}(G,P)$	$\bar{R}^1_{RI}$	$\bar{R}^1_{UI}$	$FLZE$
$Fig.2$	2.32	1.32	1.52	1.4	1.6	2
$Fig.11$	2.91	1.67	1.68	1.73	1.75	2
$Fig.15$	2	1.5	1.5	1.5	1.5	2

Figure 17: Lower and optimal achievable rate bounds

D. Fast (suboptimal) design algorithm for UI codes

Generally, the algorithms we discussed for constructing UI and RI codes are inefficient for $|V| > 10$. Here, we provide a fast (suboptimal) technique for constructing UI codes. Through the approximation of the chromatic entropy we know what is the coloring $c(X)$ of the characteristic graph that achieve the best performance. By employing Huffman coding on the random variable $c(X)$, we can construct UI codes.

At the Fig.16 we represent the achievable suboptimal UI rate with respect of instances per symbol. We see that increasing the number of instances per symbol we can achieve better performance (as we proved in theorem 4.8).

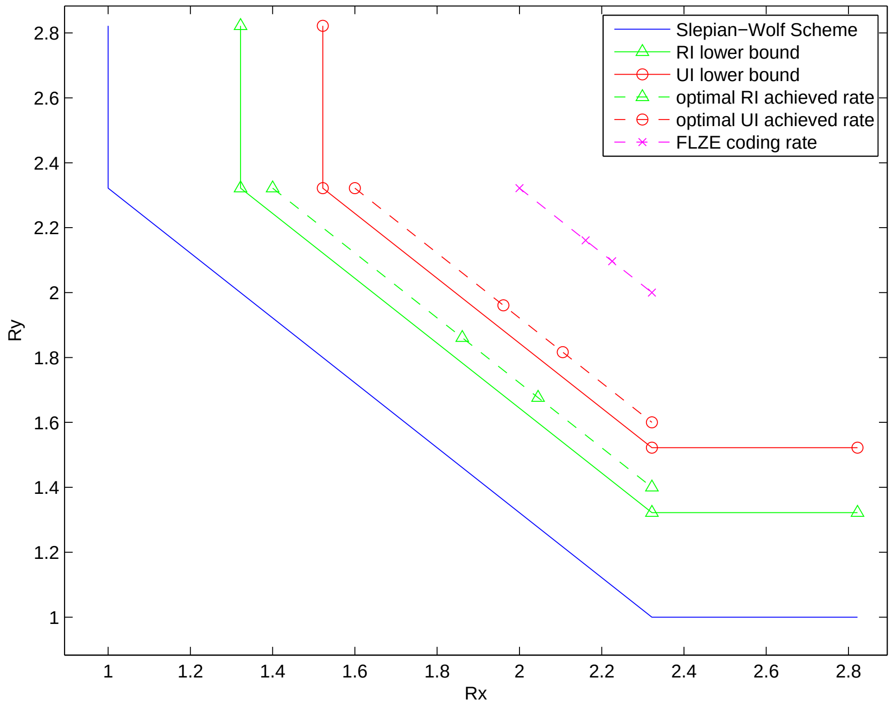

Figure 18: The optimal achievable and lower bound rates for the pentagon characteristic graph

E. Results

In Fig.17, we show the lower bounds and the optimal achievable rates for FLZE, UI and RI coding. We see that the joint encoding (either VLZE or FLZE) achieve better performance than if we encoded the sequence $\{X\}$ independently from $\{Y\}$ (in this case $R_X = H(X)$).

Finally, in Fig. 18 is illustrated the rate region for the coding techniques we discussed. We see that it is an unbounded polygon with two corners. At these points, one source is compressed at its entropy rate (side information) and can therefore be reconstructed at the decoder independently of the information received from the other source. The other one is compressed at a smaller or equal rate than its entropy. All points of the segment between two corners are achievable by time sharing. Namely, a percentage α of samples n ($\alpha \cdot n$, n is the sequence length of the source) is coded at the one point, for RI coding it is $(H(Y), H(G, P_X))$, and a fraction $(1-\alpha)$ of samples is coded at rates $(H(X), H(G, P_Y))$. This leads to the rates $R_X = \alpha H(X) + (1-\alpha)H(G, P_X)$ and $R_Y = (1-\alpha)H(Y) + \alpha H(G, P_Y)$

References

[1] N. Alon and A. Orlitsky, *Source coding and graph entropies*. IEEE Trans. Inform. Theory, vol. 42, pp. 1329-1339, Sept. 1996.

[2] M. Effros and Q. Zhao, *Lossless and near-lossless source coding for multiple access networks*. IEEE Trans. Inform. Theory, vol. 49, pp. 112-128, Jan. 2003.

[3] P. Koulgi, E.Tuncel, S. Regunathan and K. Rose, *On zero-error coding of correlated sources*. IEEE Trans. Inform. Theory, vol. 49, pp. 2856-2873 Nov. 2003.

[4] P. Koulgi, E.Tuncel, S. Regunathan and K. Rose, *On zero-error source coding with decoder side information*. IEEE Trans. Inform. Theory, vol. 49, pp. 99-111, Jan. 2003.

[5] J. Cardinal, S. Fiorini and G. Joret, *Minimum entropy combinatorial optimization problems*. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 5635 LNCS, pp. 79-88 2009.

[6] J. Cardinal, S. Fiorini and G. Joret, *Minimum entropy coloring*. J. Comb. Opt. 16(4), 361-377 2008.

[7] T. Cover and J. Thomas, *Elements of information theory*. New York: Wiley, 1991.

[8] P. Koulgi, E.Tuncel, J. Nayak and K. Rose, *On complementary graph entropy*. IEEE Trans. Inform. Theory, vol. 55, pp. 2537-2546, June 2009.

[9] N. Alon and A. Orlitsky, *A lower bound on the expected length of one-to-one codes*. IEEE Trans. Inform. Theory, vol. 40, pp. 1670-1672, Sept. 1994.

[10] P. Dragotti and M. Gastpar, *Distributed source coding: Theory, algorithms and applications*. Elsevier, Academic Press, Jan. 2009.

[11] I. Csizar and J. Körner, *Information theory, coding theorems for discrete memoryless systems*. Academic Press, New York, 1982.

[12] Z. Xiong, A. Liveris and S. Cheng, *Distributed Source Coding for Sensor Networks*. IEEE Signal Processing Magazine, Vol. 21, pp. 80-94, Sept 2004.

[13] L. Lovasz, *Combinatorial problems and exercises*. 2nd edition, Amsterdam, The Netherlands: North-Holland, 1993.

[14] J. Körner, *Coding of an information source having ambiguous alphabet and the entropy of graphs*. Trans. of the 6th Prague conference on information theory, etc, Academia, Prague, pp. 411-425, 1973.