Fernando Pujaico Rivera

Algoritmos Bit-Flipping para Decodificação Conjunta de Fontes Correlacionadas em Canais Ruidosos

Campinas
2014

Universidade Estadual de Campinas
Faculdade de Engenharia Elétrica e de Computação

Fernando Pujaico Rivera

Algoritmos Bit-Flipping para Decodificação Conjunta de Fontes Correlacionadas em Canais Ruidosos

Tese de doutorado apresentada à Faculdade de Engenharia Elétrica e de Computação como parte dos requisitos exigidos para a obtenção do título de Doutor em Engenharia Elétrica. Área de concentração: Telecomunicações e Telemática.

Orientador: Jaime Portugheis

Este exemplar corresponde à versão final da tese defendida pelo aluno, e orientado pelo Prof. Dr. Jaime Portugheis

Campinas
2014

Ficha catalográfica
Universidade Estadual de Campinas
Biblioteca da Área de Engenharia e Arquitetura
Rose Meire da Silva - CRB 8/5974

P964a

Pujaico Rivera, Fernando, 1982-
Algoritmos bit-flipping para decodificação conjunta de fontes correlacionadas em canais ruidosos / Fernando Pujaico Rivera. – Campinas, SP : [s.n.], 2014.

Orientador: Jaime Portugheis.
Tese (doutorado) – Universidade Estadual de Campinas, Faculdade de Engenharia Elétrica e de Computação.

1. Teoria da codificação. 2. Códigos de controle de erros (Teoria da informação). 3. Redes de sensores. I. Portugheis, Jaime,1959-. II. Universidade Estadual de Campinas. Faculdade de Engenharia Elétrica e de Computação. III. Título.

Informações para Biblioteca Digital

Título em outro idioma: Bit-flipping algorithms for joint decoding of correlated sources in noisy channels
Palavras-chave em inglês:
Coding theory
Error control codes (Information theory)
Sensor networks
Área de concentração: Telecomunicações e Telemática
Titulação: Doutor em Engenharia Elétrica
Banca examinadora:
Jaime Portugheis [Orientador]
Cristiano Torezzan
Marcelo Firer
Reginaldo Palazzo Júnior
Celso de Almeida
Data de defesa: 22-08-2014
Programa de Pós-Graduação: Engenharia Elétrica

iv

COMISSÃO JULGADORA - TESE DE DOUTORADO

Candidato: Fernando Pujaico Rivera

Data da Defesa: 22 de agosto de 2014

Título da Tese: "Algoritmos Bit-Flipping para Decodificação Conjunta de Fontes Correlacionadas em Canais Ruidosos"

Prof. Dr. Jaime Portugheis (Presidente): _____

Prof. Dr. Cristiano Torezzan: _____

Prof. Dr. Marcelo Firer: _____

Prof. Dr. Reginaldo Palazzo Júnior: _____

Prof. Dr. Celso de Almeida: _____

Resumo

Esta tese propõe um sistema de transmissão de informação de várias fontes correlacionadas sobre canais ruidosos. Inicialmente se aborda o problema da codificação fonte-canal conjunta. Para este problema se definirá conceitos de taxas ótimas e se apresentará a maneira de obtê-las. Posteriormente, será proposto um algoritmo de decodificação conjunta com complexidade tratável. Diversos resultados de desempenho do algoritmo proposto serão apresentados. Estes resultados mostram um bom compromisso entre desempenho e complexidade quando comparados com a decodificação independente. Finalmente, o algoritmo de decodificação conjunta seguido de uma proposta de regra de fusão é usado no problema CEO e o desempenho deste novo algoritmo é apresentado.

Palavras-chave: Redes de Sensores Distribuídos. Teorema de Slepian-Wolf. Decodificação Conjunta. Algoritmo Bit-Flipping. Problema CEO.

Abstract

This thesis proposes a system that transmits information from various correlated sources over noisy channels. Initially, it addresses the problem of joint source-channel coding. To solve this problem, concepts of optimal rates will be defined and the way to obtain them will be presented. Subsequently, a joint decoding algorithm with a tractable complexity is proposed. Several performance results of the proposed algorithm will be presented. These results show a good compromise between performance and complexity when compared to the non-joint decoding. Finally, the joint decoding algorithm, followed by a proposal for a fusion rule is used in the CEO problem and the performance of this new algorithm is presented.

Key-words: Distributed Sensor Networks. Slepian-Wolf Theorem. Joint Decoding. Bit-Flipping Algorithm. CEO Problem.

Sumário

Dedico este trabalho a todas e cada uma das pessoas que me acompanharam durante todo este tempo.

Agradecimentos

Ao meu orientador Prof. Jaime Portugheis, sou grato pela orientação.

Aos demais colegas de pós-graduação, pelas críticas e sugestões.

Este trabalho foi patrocinado pela Coordenação de Aperfeiçoamento de Pessoal de Nivel Superior (CAPES) e pela Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP) processo 2012/22641-5.

Lista de Figuras

Lista de Tabelas

Lista de Acrônimos e Notação

BF	Bit-Flipping (Algoritmo em ponto fixo de decodificação por troca de bits).
WBF	Weighted Bit-Flipping (Algoritmo em ponto flutuante, variação de BF).
PHBF	Parallel Hard Bit-Flipping (Algoritmo em ponto fixo hibrido entre BF e WBF).
LDGM	Low Density Generator Matrix (Matriz geradora de baixa densidade).
LDPC	Low Density Parity Check (Matriz de verificação de paridade de baixa densidade).

K	Número de bits não codificados.
N	Número de bits codificados.
M	Número de fontes correlacionadas.

a	Notação para amostras de uma variável aleatória (letras minúsculas do alfabeto latino).
A	Notação para variável aleatória (letras maiúsculas do alfabeto latino).
$\mathbf{A}$	Notação para matriz ou vetor (letras maiúsculas em negrito do alfabeto latino).
$\mathbf{A}'$	Apóstrofo a um vetor ou matriz, indica a operação de transposição.

U^m	Notação para a m-ésima fonte de informação.
$\hat{U}^m$	Notação da aproximação da m-ésima fonte de informação.
$\mathbf{U}^m$	Notação de um vetor da m-ésima fonte de informação.
$\mathbf{V}^m$	Notação de um vetor codificado da m-ésima fonte de informação.
$\mathbf{Z}^m$	Notação de um vetor abrupto recebido após o canal da m-ésima fonte de informação.

$H(A)$	Entropia da variável aleatória A.
$H(\mathbf{A})$	Entropia do vetor $\mathbf{A}$.
$h(p)$	Entropia binária da probabilidade p.
$Q(x)$	Função de probabilidade em x da distribuição normal padrão.
$\mathcal{I}_{\mathrm{K}}$	Matriz identidade de dimensão K .

Introdução

A transmissão eficiente da informação na forma independente de fontes correlacionadas a um nó central remoto é um problema desafiador, como é visto em (Abrardo, Ferrari & Martalo 2011), (Ferrari, Martalo, Abrardo & Raheli 2012) e (Ferrari, Martalò & Abrardo 2014). O problema torna-se ainda mais difícil quando o número de fontes é muito grande. Este é o caso de uma rede de sensores sem fio de grande escala estudado por (Barros & Tuchler 2006). Assim, é razoável assumir que os nós sensores transmitem suas observações sobre canais ortogonais com ruído, que implica que os limites teóricos podem ser atingidos com separação da codificação da fonte e canal visto em (Barros & Servetto 2006). Uma abordagem para atingir este limite pode ser usar o teorema de $Slepian - Wolf$ (Slepian & Wolf 1973a). Entretanto, o uso deste teorema pode não ser prático para redes de sensores de grande escala como é estudado em (Barros & Tuchler 2006). Outra abordagem é usar só a codificação de canal e explorar a correlação das fontes no decodificador. A decodificação de fontes correlacionadas que transmitem informação de forma codificada sobre canais ortogonais com ruído é estudada com distintos métodos. Por exemplo, em (Barros & Tuchler 2006), é usado um modelo aproximado reduzido das dependências entre as fontes, que é representado mediante grafo-fatores. Os grafo-fatores são utilizados para decodificar de forma conjunta mediante o algoritmo soma-produto. Além disso, se uma fonte é considerada a original e as outras versões ruidosas dela, o cenário pode ser considerado como um caso de decodificação com múltipla informação lateral como em (Shamai & Verdu 1995).

Nesta tese, adota-se esta posterior abordagem de usar só codificação de canal além de um cenário com um codificador de complexidade controlável. A fim de lidar com esta complexidade propõe-se um algoritmo para a decodificação de múltiplas fontes correlacionadas, baseado no algoritmo de decodificação $Bit - Flipping$ proposto em (Gallager 1962). Um modelo de correlação simples foi adotado para obter o desempenho do algoritmo proposto. Os modelos adotados são similares aos descritos em (Haghighat, Behroozi & Plant 2008),(K. Kobayashi & Katayama 2009),(Ferrari et al. 2012) e (Ferrari et al. 2014). De maneira distinta da literatura, nesta tese se consideram cenários com um grande número de fontes e com valores de correlação do modelo gerados aleatoriamente. Também é obtida uma taxa ótima para distintos cenários com um número moderado de fontes.

O problema de transmitir informação de fontes correlacionadas para um nó central remoto pode ser adaptado e interpretado como o problema CEO (do inglês, Chief Executive Officer).

Nessa linha em (Haghighat et al. 2008) e (Ferrari et al. 2014) se faz uma decodificação conjunta em duas etapas sobre um grupo de fontes correlacionadas geradas a partir de uma fonte escondida. Primeiro se faz uma decodificação conjunta usando o algoritmo $BCJR$, obtendo para cada vetor da fonte (não codificada) um vetor de log-verossimilhança a posteriori. Logo numa segunda etapa é usado este vetor para obter outro vetor com uma estimação da fonte escondida.

Nesta tese o modelo de decodificação para o problema CEO é também em duas etapas. A primeira é realizada mediante um algoritmo de decodificação conjunta da família dos algoritmos $Bit-Flipping$, para obter uma probabilidade de erro na transmissão de informação no canal. Na segunda etapa se propõe um algoritmo CEO que usa as probabilidades de erro de canal antes obtidas e as probabilidade do modelo de geração de fontes correlacionadas para estimar a informação enviada pela fonte escondida.

A seguinte notação é usada neste trabalho. As variáveis são denotadas com letras maiúsculas ou minúsculas (e.g., X x), as variáveis aleatórias são denotadas por letras maiúsculas cursivas (e.g., X) e seus valores (amostras) por letras minúsculas cursivas (e.g., x). No caso de variáveis aleatórias e amostras diferentes mas do mesmo tipo, estas serão denotadas com um super índice (e.g., X^m e x^m). Vetores e matrizes são denotados por letras maiúsculas em negrito (e.g., $\mathbf{X}$). Elementos dos vetores e matrizes são denotados com sub índices, em maiúsculas em caso de vetores e matrizes de variáveis aleatórias e em minúsculas no caso de variáveis (e.g., X_i e x_i).

Contribuições da Tese

- No Capítulo 2 apresenta-se três contribuições. A primeira é a obtenção de uma forma de cálculo da taxa de codificação fonte-canal ótima em termos de uso de energia do codificador. Para este fim se define um sistema de transmissão de informação com várias fontes correlacionadas, onde todas as fontes usam a mesma taxa de codificação. A segunda contribuição é a apresentação de uma expressão para obter valores ótimos das taxas de codificação fonte-canal, no caso em que cada codificador usa um valor distinto para esta taxa. Novamente se otimiza em termos de energia utilizada pelo codificador. A terceira contribuição é a definição de uma variante do limite de Shannon para a capacidade do canal, no caso de sistemas com fontes correlacionadas. Esta variante do limite é chamada nesta tese limite virtual de Shannon.

- No Capítulo 3 apresenta-se duas contribuições. A primeira é a definição de uma variante simplificada do algoritmo "Parallel Weighted Bit-Flipping". A segunda contribuição é a definição de um algoritmo para a decodificação conjunta em sistemas de transmissão com múltiplas fontes quando estas possuem correlação. Este algoritmo de decodificação conjunta está baseado no algoritmo de decodificação "Bit-Flipping" e tem duas variantes: a primeira variante está inspirada no algoritmo "Parallel Hard Bit-Flipping" e a segunda variante no algoritmo "Parallel Weighted Bit-Flipping" simplificado. Este algoritmo de decodificação conjunta é o ponto de estudo principal desta tese.

- O Capítulo 4 define uma regra de decisão para a predição da fonte escondida e será aplicada

no problema CEO.

Capítulo 1

Conceitos Básicos

Neste capítulo mostra-se o fundamento teórico necessário para o estudo ordenado desta tese. Dado que o ponto de estudo principal foca-se no algoritmo de decodificação conjunta, é importante descrever qual é o modelo de sistema de transmissão usado na decodificação. O algoritmo de decodificação conjunta aplica-se sobre os dados do envio da informação codificada de M fontes correlacionadas U^m, $m \in \{1, 2, ..., \mathrm{M}\}$, através de M canais gaussianos. As fontes correlacionadas são também chamadas de fontes distribuídas e podem ser vistas como imagens independentes de uma cena em comum. Assim, a correlação entre elas pode tomar qualquer valor, inclusive correlação nula.

Cada uma das M fontes distribuídas, U^1, U^2, ..., U^{M}, recebe uma codificação de fonte para obter taxas de informação, R_1, R_2, ..., R_{M}, respectivamente, de modo que as taxas sejam menores que as taxas de informação mínimas atingidas por uma codificação independente de fontes não distribuídas. A escolha das taxas segue o teorema de $Slepian - Wolf$ estudado em (Slepian & Wolf 1973b, Cover & Thomas 2006), ver Seção 1.2. Após a codificação de fonte poderá ocorrer uma codificação de canal antes do envio das informações para o destino.

1.1 Tipos de Codificação de Fonte

Dado um conjunto de M fontes U^m, $m \in \{1, 2, ..., \mathrm{M}\}$, a codificação das fontes dependerá do fato destas fontes estarem ou não distribuídas e do fato da codificação entre elas ser feita de forma independente ou dependente, como pode ver-se na Figura 1.1.

Em (Pradhan & Ramchandran 2003) chama-se de fontes distribuídas a um conjunto de fontes que entregam informação correlacionada, de modo que a informação das fontes representam amostras individuais de um mesmo fenômeno. Isto é, a informação das fontes tem um grau de correlação. O limite de compressão da informação para as fontes distribuídas está governado pelo teorema de $Slepian - Wolf$. No caso das fontes não distribuídas estas são fontes com correlação zero. O limite de compressão da informação para as fontes não distribuídas é a entropia de cada fonte, $H(U^m)$. Definições de entropia, entropia condicional e entropia conjunta podem ser encontradas em (Cover & Thomas 2006).

Chama-se codificação dependente de fontes distribuídas, quando a informação de todas as fontes é conhecida pelo codificador ou codificadores. Este tipo de codificação é interessante

para modelos de fontes com comunicação entre elas. Por outro lado a codificação independente de fontes distribuídas usa só a informação de cada fonte para codificar a informação. A codificação independente é interessante para fontes sem comunicação entre elas, ou para fontes não distribuídas.

Assim podemos ver três tipos básicos de codificação de fonte, seguindo a Figura 1.1.

1. Codificação dependente de fontes distribuídas: quando se trata de fontes distribuídas com comunicação entre elas.

2. Codificação independente de fontes distribuídas: quando se trata de fontes distribuídas sem comunicação entre elas. Este tipo de codificação será adotado nesta tese.

3. Codificação de fonte: quando se trata de fontes não distribuídas com ou sem comunicação entre elas e codificação independente.

Também é possível fazer a codificação de fonte junto com a codificação de canal para os três tipos de codificação antes mencionados.

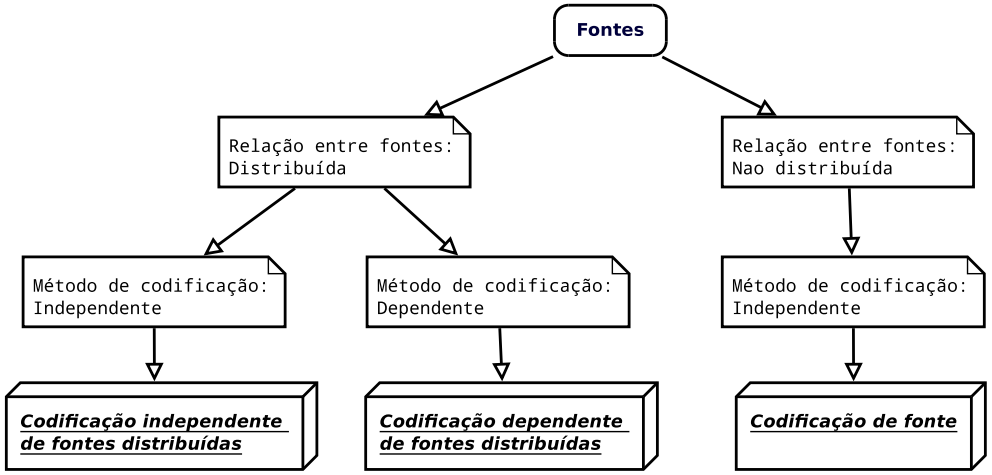

Figura 1.1: Tipos de codificação de fonte.

1.2 Teorema de Slepian-Wolf

Seguindo o teorema de $Slepian - Wolf$ (Slepian & Wolf 1973b), sabe-se que dadas duas fontes de informações correlacionadas U^1 e U^2, são codificadas de formas independentes, de modo que a fonte U^1 tenha uma taxa de informação R_1 na saída do codificador de fonte e a fonte U^2 uma taxa R_2, como na Figura 1.2. As taxas de informações mínimas e necessárias a serem enviadas por dois canais sem ruído de modo que seja possível obter uma decodificação conjunta de U^1 e U^2 sem perda de informação, são dadas pelas seguintes equações:

$$\mathrm{R}_1 \geq H(U^1|U^2), \tag{1.1}$$

$$\mathrm{R}_2 \geq H(U^2|U^1), \tag{1.2}$$

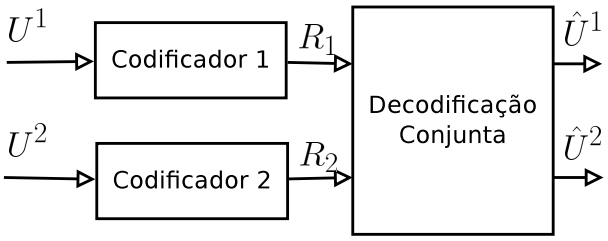

Figura 1.2: Codificação de duas fontes estatisticamente dependentes.

$$\mathrm{R}_1 + \mathrm{R}_2 \geq H(U^1U^2). \tag{1.3}$$

Outra forma de entender estas equações, é notando que (1.1) e (1.2) descrevem que o mínimo a ser enviado por cada canal é a informação que é exclusiva desse canal, pois não existe redundância nesses dados, ver Figura 1.3. Mas a equação (1.3) mostra que: tendo garantido que a informação exclusiva de cada canal é enviada, a soma das informações enviadas pelos dois canais tem que atingir como mínimo a entropia conjunta de U^1 e U^2. Considerando as equações (1.1), (1.2) e (1.3) a região atingível pelo par de taxas $(\mathrm{R}_1, \mathrm{R}_2)$ está representada pela Figura 1.4.

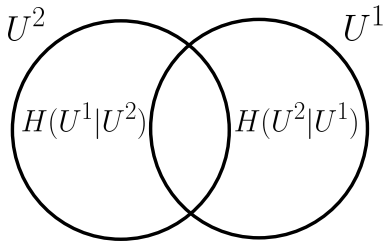

Figura 1.3: Duas fontes correlacionadas.

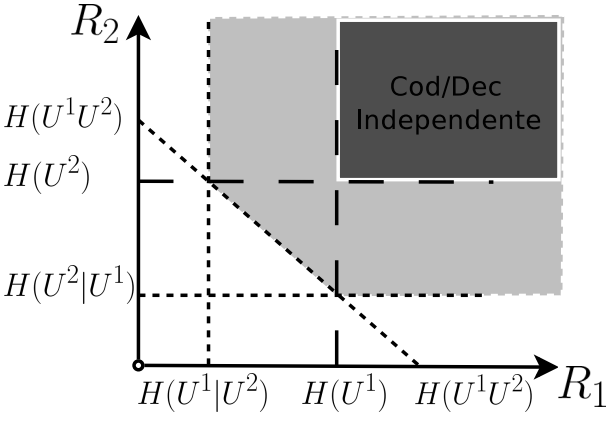

Figura 1.4: Região atingível pelo par de taxas $(\mathrm{R}_1, \mathrm{R}_2)$.

1.2.1 Teorema de Slepian-Wolf de Três Fontes

Para o caso em que se tem três fontes correlacionadas U^1,U^2 e U^3 como o caso da Figura 1.5, o teorema de $Slepian-Wolf$ para as mínimas taxas de informação codificada R_1, R_2 e R_3,

pode ser expressado pelas seguintes equações:

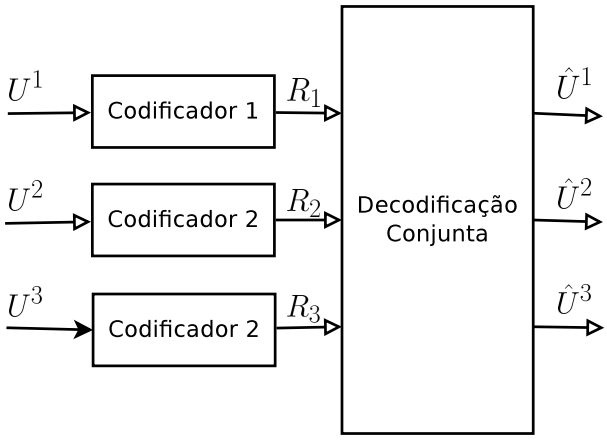

Figura 1.5: Codificação de três fontes estatisticamente dependentes.

$$\mathrm{R}_1 \geq H(U^1|U^2U^3), \tag{1.4}$$

$$\mathrm{R}_2 \geq H(U^2|U^1U^3), \tag{1.5}$$

$$\mathrm{R}_3 \geq H(U^3|U^1U^2), \tag{1.6}$$

$$\mathrm{R}_1 + \mathrm{R}_2 \geq H(U^1U^2|U^3), \tag{1.7}$$

$$\mathrm{R}_1 + \mathrm{R}_3 \geq H(U^1U^3|U^2), \tag{1.8}$$

$$\mathrm{R}_2 + \mathrm{R}_3 \geq H(U^2U^3|U^1), \tag{1.9}$$

$$\mathrm{R}_1 + \mathrm{R}_2 + \mathrm{R}_3 \geq H(U^1U^2U^3). \tag{1.10}$$

Da mesma forma que o modelo para duas fontes correlacionadas, quando se tem três fontes o primeiro grupo de equações, formada por (1.4),(1.5) e (1.6), indica que cada fonte pelo menos tem que enviar a informação que elas possuam de forma exclusiva. O segundo grupo de equações, formado por (1.7), (1.8) e (1.9), expressa que a soma da informação de dois canais qualquer tem que conter pelo menos a informação que é exclusiva deles. Por último a equação (1.10) expressa que a soma da informação de três canais tem que enviar pelo menos a informação exclusiva deles, neste caso a entropia conjunta $H(U^1U^2U^3)$.

Exemplo 1.2.1 *Três fontes de informações correlacionadas U^1, U^2 e U^3, com taxas de informação R_1, R_2 e R_3, e entropias condicionadas $H(U^1|U^2U^3) = H(U^2|U^1U^3) = H(U^3|U^1U^2) = 0.589327$, $H(U^1U^2|U^3) = H(U^2U^3|U^1) = H(U^3U^1|U^2) = 1.269404$ e $H(U^1U^2U^3) = 2.269404$, gerarão um gráfico para as taxas de informações mínimas como mostra a Figura 1.6.*

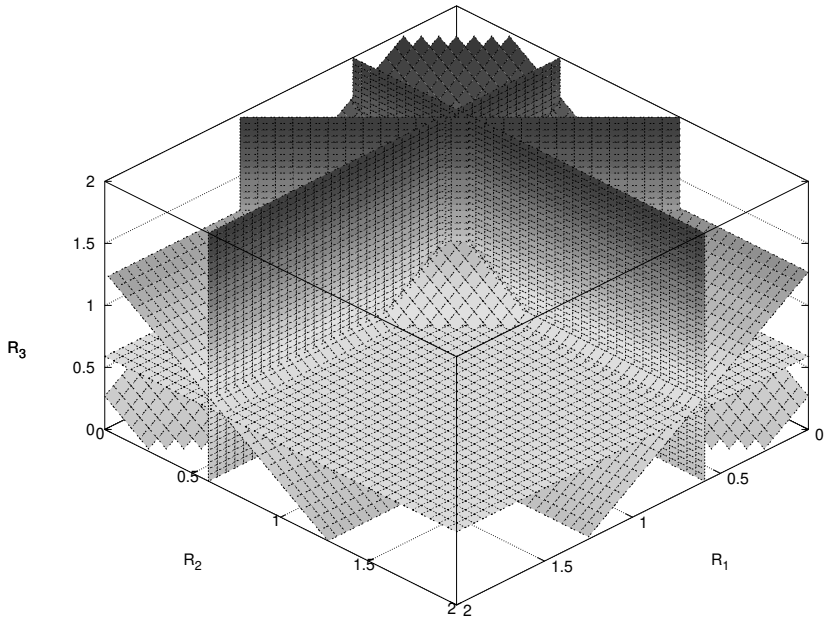

Figura 1.6: Slepian-Wolf 3D.

1.2.2 Teorema de Slepian-Wolf de M Fontes

Seguindo o teorema de $Slepian-Wolf$ descrito em (Slepian & Wolf 1973b, Cover & Thomas 2006), se sabe que se temos M fontes correlacionadas e identicamente distribuídas $\{U^1, U^2, ..., U^{\mathrm{M}}\}$. As taxas de informação atingíveis, para que a informação conjunta delas seja recuperável, são $\{\mathrm{R}_1, \mathrm{R}_2, ..., \mathrm{R}_{\mathrm{M}}\}$. Define-se $S \subseteq \{1, 2, ..., \mathrm{M}\}$, S^c como seu complemento e

$$R(S) \equiv \sum_{m \in S} \mathrm{R}_m, \tag{1.11}$$

$$U(S) \equiv \bigcup_{m \in S} U^m. \tag{1.12}$$

O teorema de $Slepian - Wolf$ afirma que $\forall S \subseteq \{1, 2, ..., \mathrm{M}\}$ é valido que

$$R(S) \geq H(U(S)|U(S^c)). \tag{1.13}$$

Considerando que cada subconjunto S tem t elementos, o número de equações que geram os subconjuntos S de t elementos é $\binom{\mathrm{M}}{t}$. O teorema de $Slepian - Wolf$ para M fontes gera no total T_{eq} equações, onde

$$T_{eq} = \sum_{t=1}^{\mathrm{M}} \binom{\mathrm{M}}{t} = 2^{\mathrm{M}} - 1. \tag{1.14}$$

Por exemplo, para o caso em que M = 100 o teorema de $Slepian - Wolf$ contará com $1.26765\ 10^{30}$ equações de restrição.

Outro dado importante é que cada taxa de informação R_m, $\forall m \in \{1, 2, ..., m\}$; obtida após a codificação de fonte; deve ser sempre menor ou igual que a informação $H(U^m)$, ou seja, $\mathrm{R}_m \leq H(U^m)$. Em geral pode-se definir que:

$$H(S) \equiv \sum_{m \in S} H(U^m). \tag{1.15}$$

Portanto

$$H(S) \geq R(S) \geq H(U(S)|U(S^c)). \tag{1.16}$$

1.3 Modelos de Canais

Os modelos de canais neste trabalho distinguem-se pela forma como o ruído é introduzido e o tipo de dados que aparecem na sua saída.

1.3.1 Canal BSC

Um canal binário simétrico ou BSC (do inglês Binary Symmetric Channel) é um canal de comunicação com entrada binária de modo que na saída obtém-se uma versão ruidosa da entrada com uma probabilidade de erro p, como ilustrado na Figura 1.7. Se o bit enviado pelo canal é 0 então tem-se uma probabilidade $1 - p$ de receber um bit 0 na sua saída e uma probabilidade p de receber um bit 1. Agora se é enviado um 1, se tem uma probabilidade $1 - p$ de receber um bit 1 na sua saída e uma probabilidade p de receber um bit 0.

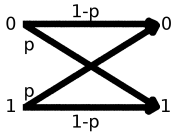

Figura 1.7: Canal BSC.

Se definirmos uma variável aleatória X na entrada do canal BSC e uma variável aleatória Y na saída, obtemos as seguintes relações:

$$P(X \neq Y|X) = p, \tag{1.17}$$

$$P(X = Y|X) = 1 - p. \tag{1.18}$$

Em forma geral a capacidade do canal para um canal com entrada X e saída Y é dada pela seguinte equação:

$$C = \max_{P(X)}\{I(X, Y)\}, \tag{1.19}$$

onde $I(X, Y) = H(Y) - H(Y|X)$ é a informação mutua entre as variáveis X e Y. Do anterior deduz-se que

$$C = \max_{P(X)}\{H(Y) - h(p)\}, \tag{1.20}$$

$$C_{BSC} = 1 - h(p). \tag{1.21}$$

Isso acontece quando a fonte X tem uma probabilidade $P(X = 0) = 0.5$. A equação (1.21) mostra que a capacidade do canal BSC é 1 bit diminuído pela quantidade de informação do erro $h(p)$ no canal, onde $h(p)$ é a entropia binária definida como:

$$h(p) \equiv -p\ log_2(p) - (1-p)log_2(1-p). \tag{1.22}$$

1.3.2 Canal BI-AWGN

Um canal de ruído gaussiano aditivo branco ou AWGN (do inglês Additive White Gaussian Noise) é um canal de comunicações contínuo onde o ruído é adicionado linearmente à informação transmitida. Se definimos a entrada do canal como uma variável aleatória X, a saída do canal como Y e ao ruído gaussiano adicionado como W, a equação que descreve as relações é a seguinte

$$Y = X + W, \tag{1.23}$$

onde W tem a seguinte função de densidade de probabilidade,

$$f_W(w) \equiv f_W(W = w) = \frac{1}{\sqrt{2\pi\sigma^2}}\ exp[-\frac{w^2}{2\sigma^2}]. \tag{1.24}$$

Uma variação do canal $AWGN$ é o canal $BI - AWGN$ (do inglês, Binary Input $AWGN$). Este é um canal com ruído branco aditivo e gaussiano, mas só com dois valores de entrada X, isto é, tem uma entrada binária, $\{-1, +1\}$.

As probabilidades de transição deste canal são dadas por:

$$p(y|x) \equiv P(Y = y|X = x) = \frac{1}{\sqrt{2\pi}\sigma}e^{-\frac{(y-x)^2}{2\sigma^2}}. \tag{1.25}$$

A probabilidade da saída Y é:

$$P(y) = P(y|x = +1)P(x = +1) + P(y|x = -1)P(x = -1). \tag{1.26}$$

Se os símbolos de entrada são equiprováveis, isto é, $P(x = -1) = P(x = +1) = 0.5$, então

$$p(y) = \frac{1}{2}\{P(y|x = +1) + P(y|x = -1)\}. \tag{1.27}$$

Neste caso, a capacidade $\mathrm{C}_{BI-AWGN}$ do canal $BI\text{-}AWGN$, pode ser calculada como segue:

$$\mathrm{C}_{BI-AWGN} = \phi(\sigma^2) = \frac{1}{2}\sum_{x=\pm 1}\int_{-\infty}^{+\infty} p(y|x)log_2(\frac{p(y|x)}{p(y)})dy \tag{1.28}$$

$$= -\int_{-\infty}^{+\infty} p(y)log_2(p(y))dy - 0.5log_2(2\pi e\sigma^2). \tag{1.29}$$

Note que a energia por símbolo foi normalizada para a unidade, isto é, $E_s = 1$, e assim $p(y)$ e $p(y|x)$ dependem apenas da variância σ^2.

É necessário chegar a uma relação entre E_b/N_0 e σ^2. Pode-se deduzir que o valor quadrático meio do ruído é $E_W = E[W^2] = \sigma^2$. E dada uma densidade espectral de potência constante com valor $S_W(f) = N_0/2$,

$$E_W = \lim_{B\to\infty} \frac{1}{2B} \int_{-B}^{B} (N_0/2) df = N_0/2. \tag{1.30}$$

Assim se obtém a relação $\sigma^2 = N_0/2$. Por outro lado se a informação que vai percorrer o canal é codificada, a energia usada em cada símbolo enviado E_s é dada por:

$$E_b = \frac{\mathrm{N}}{\mathrm{K}} E_s, \tag{1.31}$$

sendo E_b a energia usada para transmitir um bit de informação, K o comprimento do bloco de informação e N o comprimento da palavra codificada. Esta expressão é fácil de deduzir pensando que a energia E_b para transmitir um bit de informação num canal não codificado é $E_b = E_s$ e a energia $\mathrm{K}E_b$ para transmitir K bits de informação num canal codificado é $\mathrm{K}E_b = \mathrm{N}E_s$. Assim é mais custoso, em termos de energia, transmitir um bit de informação num canal codificado. De (1.31), (1.30), lembrando que $E_s \equiv 1$ e que $r = \mathrm{K/N}$ obtemos:

$$E_b/N_0 = 1/(2r\sigma^2). \tag{1.32}$$

Assim, para uma capacidade de canal $\mathrm{C}_{BI-AWGN}$ fixa, r e E_b/N_0 seguem a seguinte relação

$$\mathrm{C}_{BI-AWGN} = \phi(\frac{1}{2rE_b/N_0}). \tag{1.33}$$

1.3.3 Canal BI-AWGN com Decisão Abrupta

O canal *BI-AWGN* com decisão abrupta (do inglês, "hard-decision *BI-AWGN*") é um canal *BI-AWGN* em cuja saída Y se aplica uma regra de decisão que transforma este valor real em Z, que só pode ter dois valores (saída binária).

$$Z = \begin{cases} 0 & Y \leq 0 \\ 1 & Y > 0 \end{cases}. \tag{1.34}$$

Dado que um canal *BI-AWGN* com decisão abrupta tem uma entrada binária é uma saída binária, este tipo de canal pode ser representado, ou ter como equivalente, um canal *BSC*. É conhecido (Cover & Thomas 2006) que a relação entre a probabilidade de erro de bit p do canal *BSC* e o valor de sinal e ruído E_b/N_0 num canal codificado de tipo *BI-AWGN* com decisão abrupta é dado por:

$$p = Q(\sqrt{2rE_b/N_0}). \tag{1.35}$$

onde $Q(.)$ é chamado de "Q-function" e é definido como:

$$Q(x) = \frac{1}{\sqrt{2\pi}} \int_x^\infty \exp\left(-\frac{u^2}{2}\right) du. \tag{1.36}$$

Dado que a capacidade de um canal *BSC* é $\mathrm{C}_{BSC} = 1 - h(p)$, onde $h(.)$ é a entropia binária, podemos deduzir que a capacidade de um canal *BI-AWGN* com decisão abrupta é

$$\mathrm{C}_{HD-BI-AWGN} = 1 - h(Q(\sqrt{2E_s/N_0})), \tag{1.37}$$

$$\mathrm{C}_{HD-BI-AWGN} = 1 - h(Q(\sqrt{2rE_b/N_0})). \tag{1.38}$$

1.4 Códigos Corretores de Erros

Um código corretor de erros é uma regra para atribuir a cada palavra de informação **U** uma única palavra código **V** que contém redundância. Isto é importante porque ante a presença de ruído a palavra de informação **U** pode distorcer-se e ser confundida com alguma outra palavra de informação **U**. No novo mapeamento a palavra código **V** de **U** está mais espaçada com respeito as outras palavras código correspondentes a **U**. Por exemplo, para o caso de códigos binários se mapeia uma palavra de informação de K bits em uma palavra código de N bits, como pode-se ver na Figura 1.8.

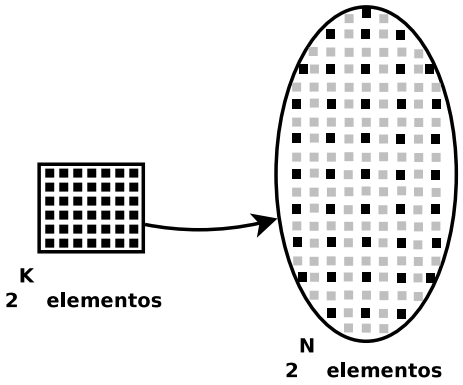

Figura 1.8: Código corretor de erros.

Nos métodos de mapeamento temos os códigos de bloco lineares para elementos binários, este mapeamento é feito através de uma matriz geradora **G**. Considerando a palavra de informação

$$\mathbf{U} = \begin{pmatrix} u_0 & u_1 & u_2 & ... & u_{\mathrm{K}-1} \end{pmatrix}, \tag{1.39}$$

e a palavra código

$$\mathbf{V} = \begin{pmatrix} v_0 & v_1 & v_2 & ... & v_{\mathrm{N}-1} \end{pmatrix}. \tag{1.40}$$

Então,

$$\mathbf{V} = \mathbf{U}\ \mathbf{G}, \tag{1.41}$$

onde

$$\mathbf{G} = \begin{pmatrix} \mathbf{g}_0 \\ \mathbf{g}_1 \\ \mathbf{g}_2 \\ \vdots \\ \mathbf{g}_{\mathrm{K}-1} \end{pmatrix} = \begin{pmatrix} g_{00} & g_{01} & g_{02} & \cdots & g_{0(\mathrm{N}-1)} \\ g_{10} & g_{11} & g_{12} & \cdots & g_{1(\mathrm{N}-1)} \\ g_{20} & g_{21} & g_{22} & \cdots & g_{2(\mathrm{N}-1)} \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ g_{(\mathrm{K}-1)0} & g_{(\mathrm{K}-1)1} & g_{(\mathrm{K}-1)2} & \cdots & g_{(\mathrm{K}-1)(\mathrm{N}-1)} \end{pmatrix}. \tag{1.42}$$

Na matriz geradora **G**, cada elemento $g_{kn} \in GF(2)$, $\forall k \in \{0, 1, ..., \mathrm{K}-1\}$ e $\forall n \in \{0, 1, ..., \mathrm{N}-1\}$. Onde $GF(2)$ é um corpo de Galois de dois elementos. Cada uma das linhas de **G** são $\mathrm{N}-uplas$ linearmente independentes. O vetor **V** é um elemento no espaço vetorial formado pelos vetores $\mathbf{g}_k$, isto é,

$$\mathbf{V} = u_0\mathbf{g}_0 + u_1\mathbf{g}_1 + u_2\mathbf{g}_2 + ... + u_{\mathrm{K}-1}\mathbf{g}_{\mathrm{K}-1}. \tag{1.43}$$

O conjunto de todos os pontos $\mathbf{V}$ formam um subespaço vetorial de dimensão K do espaço vetorial de dimensão N formado por todas as $\mathrm{N}-uplas$ com elementos em $GF(2)$, também chamado $EG(\mathrm{N},2)$.

Dado que os vetores $\mathbf{g}_k$ são linearmente independentes, se tem uma relação unívoca entre um ponto $\mathbf{U}$ e um ponto $\mathbf{V}$, pelo qual é possível a decodificação do $\mathbf{V}$ para obter novamente $\mathbf{U}$. O hiperplano $\mathbf{F} = \{\mathbf{V} : \mathbf{V} = \mathbf{UG}\}$ de dimensão K, tem em $EG(\mathrm{N},2)$ $\mathrm{N}-\mathrm{K}$ vetores linearmente independentes e ortogonal a cada ponto do plano $\mathbf{F}$, isto é, cada palavra código $\mathbf{V}$ e os vetores $\mathbf{g}_k$ são ortogonais a estes $\mathrm{N}-\mathrm{K}$ vetores. Se chamará de $\mathbf{h}_l \in EG(\mathrm{N},2)$, $\forall l \in \{0,1,...,\mathrm{N}-\mathrm{K}-1\}$ a cada um desses vetores. Com $\mathbf{h}_l$ se constrói a matriz $\mathbf{H}$,

$$\mathbf{H} = \begin{pmatrix} \mathbf{h}_0 \\ \mathbf{h}_1 \\ \mathbf{h}_2 \\ \vdots \\ \mathbf{h}_{\mathrm{N}-\mathrm{K}-1} \end{pmatrix} = \begin{pmatrix} h_{00} & h_{01} & h_{02} & \dots & h_{0(\mathrm{N}-1)} \\ h_{10} & h_{11} & h_{12} & \dots & h_{1(\mathrm{N}-1)} \\ h_{20} & h_{21} & H_{22} & \dots & h_{2(\mathrm{N}-1)} \\ \vdots & \vdots & \vdots & \dots & \vdots \\ h_{(\mathrm{N}-\mathrm{K}-1)0} & h_{(\mathrm{N}-\mathrm{K}-1)1} & h_{(\mathrm{N}-\mathrm{K}-1)2} & \dots & h_{(\mathrm{N}-\mathrm{K}-1)(\mathrm{N}-1)} \end{pmatrix}. \tag{1.44}$$

Como a matriz $\mathbf{H}$ de $\mathrm{N}-\mathrm{K}$ linhas e N colunas é ortogonal a $\mathbf{V}$ e $\mathbf{G}$, $\mathbf{H}$ pode ser usado como uma medida de verificação de que um ponto $\mathbf{Z}$ qualquer pertence ou não ao subespaço $\mathbf{F}$. Primeiro calcula-se $\mathbf{S}$ com

$$\mathbf{ZH}^t = \mathbf{S}, \tag{1.45}$$

e se $\mathbf{S} = \mathbf{0}$ se sabe que $\mathbf{Z}$ pertence a $\mathbf{F}$. É possível determinar $\mathbf{H}$ a partir de $\mathbf{G}$, procurando seu espaço nulo, como segue,

$$\mathbf{GH}^t = 0. \tag{1.46}$$

Nas equações (1.45) e (1.46) o símbolo t representa a transposta da matriz. O uso do vetor $\mathbf{S}$ para estimar os bits errados do vetor $\mathbf{Z}$ será estudado no Capítulo 3.

1.4.1 Forma Sistemática e Códigos *LDGM*

Uma forma de construir a matriz $\mathbf{G}$, que facilita a obtenção do vetor $\mathbf{U}$ a partir do vetor $\mathbf{V}$, é fazendo com que dentro da palavra código $\mathbf{V}$ esteja incorporado o vetor de informação $\mathbf{U}$, e o restante da palavra é completado com os bits de paridade, ver Figura 1.9. As matrizes geradoras $\mathbf{G}$ deste tipo se chamam de *LDGM* (do inglês "Low Density Generator Matrix") por sua baixa densidade de "1"s.

Para obter isto, usa-se uma matriz identidade $\mathcal{I}_{\mathrm{K}}$ de dimensão K e uma matriz de paridade $\mathbf{P}$ de K linhas e $\mathrm{N}-\mathrm{K}$ colunas dado por:

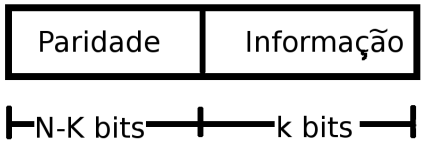

Figura 1.9: Código sistemático.

$$\mathbf{P} = \begin{pmatrix} p_{00} & p_{01} & p_{02} & \cdots & p_{0(\mathrm{N-K-1})} \\ p_{10} & p_{11} & p_{12} & \cdots & p_{1(\mathrm{N-K-1})} \\ p_{20} & p_{21} & p_{22} & \cdots & p_{2(\mathrm{N-K-1})} \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ p_{(\mathrm{K-1})0} & p_{(\mathrm{K-1})1} & p_{(\mathrm{K-1})2} & \cdots & p_{(\mathrm{K-1})(\mathrm{N-K-1})} \end{pmatrix}. \tag{1.47}$$

Assim, escrevemos $\mathbf{G}$ como

$$\mathbf{G} = [\mathbf{P}\ \mathcal{I}_{\mathrm{K}}]. \tag{1.48}$$

Para obter a palavra código $\mathbf{V}$ usa-se a equação (1.41). Outra vantagem de construir a matriz $\mathbf{G}$ de forma sistemática é que é fácil obter a matriz $\mathbf{H}$, que é ortogonal a $\mathbf{G}$ como pode-se ver na equação (1.46). Então $\mathbf{H}$ é dado por

$$H = \left[\mathcal{I}_{\mathrm{N-K}}\ \mathbf{P}^t\right]. \tag{1.49}$$

No artigo (Garcia-Frias & Zhong 2003) foi descrito um limitante inferior para a taxa de erro de bit BER (do inglês, Bit Error Rate), em matrizes de verificação de paridade de tipo $LDGM$ quando a matriz $\mathbf{P}$ tem uma quantidade constante de "1"s por linha. Designando a quantidade de uns por linha de $\mathbf{P}$ por $\mathcal{X}$. Assim se $\mathcal{X}$ é impar o BER inferior é

$$BER_{Limite\ Impar}(\mathcal{X}, p) \approx \sum_{k=(\mathcal{X}+1)/2}^{\mathcal{X}} \binom{\mathcal{X}}{k} p^k (1-p)^{\mathcal{X}-k}, \tag{1.50}$$

enquanto que no caso de ter $\mathcal{X}$ par o BER inferior é

$$BER_{Limite\ Par}(\mathcal{X}, p) \approx \sum_{k=(\mathcal{X}/2)+1}^{\mathcal{X}} \binom{\mathcal{X}}{k} p^k (1-p)^{\mathcal{X}-k} + p\binom{\mathcal{X}}{\mathcal{X}/2} p^{\mathcal{X}/2} (1-p)^{\mathcal{X}/2}, \tag{1.51}$$

onde p é a probabilidade de erro de bit do canal. É interessante ressaltar que o BER limite correspondente a $\mathcal{X}$ par é igual ao BER correspondente do $\mathcal{X}$ impar imediato superior. Isto é, $BER_{Limite\ Par}(2a, p) = BER_{Limite\ Impar}(2a+1, p)$, para todo a inteiro.

1.4.2 Códigos LDPC

Os códigos $LDPC$, do inglês "Low-Density Parity-Check", foram apresentados por Gallager em (Gallager 1962). A importância deste tipo de códigos não se fez evidente até que foram

redescobertos por MacKay em (MacKay 1999). Estes códigos são caracterizados por ter matrizes de verificação de paridade **H** com uma baixa quantidade de "1"s na sua composição. Entre as matrizes de verificação de paridade **H** de tipo $LDPC$ existem dois tipos: As chamadas matrizes irregulares e regulares. Uma matriz é chamada de regular quando tem uma quantidade d_v fixa de "1"s por coluna e uma quantidade d_c fixa de "1"s por linha, no caso contrário a matriz é chamada de irregular. É fácil notar que em uma matriz regular, pode-se calcular a quantidade total de "1"s na matriz **H** da seguinte forma:

$$d_v \mathrm{N} = d_c(\mathrm{N} - \mathrm{K}). \tag{1.52}$$

Pode-se expressar a taxa do código r em função de (d_v, d_c) para matrizes de verificação de tipo $LDPC$ regular como

$$r = 1 - \frac{d_v}{d_c} = 1 - \frac{\mathrm{N} - \mathrm{K}}{\mathrm{N}} = \frac{\mathrm{K}}{\mathrm{N}}. \tag{1.53}$$

1.4.3 Relação Entre os Códigos do Tipo $LDPC$ e a Forma Sistemática

Imagine-se o caso em que se tem uma matriz binária **H** de tipo $LDPC$ com L linhas e N colunas. Esta pode ser usada como matriz de verificação de paridade. A pergunta seria: Qual é a matriz geradora **G** para esta matriz?. Para resolver isto é necessário saber que é válido que:

$$\hat{\mathbf{G}}\hat{\mathbf{H}}^t = 0, \tag{1.54}$$

onde $\hat{\mathbf{G}}$ e $\hat{\mathbf{H}}$ são matrizes geradas a partir de combinações lineares das linhas de **G** e **H** respectivamente. Isto é, que qualquer combinação linear de **H** é ortogonal a **G** e vice-versa. Assim fazendo combinações lineares das linhas de **H** (método de Gauss) é possível obter uma matriz $\hat{\mathbf{H}}$ da forma

$$\hat{H} = \begin{pmatrix} 1 & 0 & 0 & \dots & 0 & p_{00} & p_{10} & p_{20} & \cdots & p_{(K-1)0} \\ 0 & 1 & 0 & \dots & 0 & p_{01} & p_{11} & p_{21} & \cdots & p_{(K-1)1} \\ 0 & 0 & 1 & \dots & 0 & p_{02} & p_{12} & p_{22} & \cdots & p_{(K-1)2} \\ \vdots & \vdots & \vdots & \dots & \vdots & \vdots & \vdots & \vdots & \cdots & \vdots \\ 0 & 0 & 0 & \dots & 1 & p_{0(N-K-1)} & p_{1(N-K-1)} & p_{2(N-K-1)} & \cdots & p_{(K-1)(N-K-1)} \\ 0 & 0 & 0 & \dots & 0 & 0 & 0 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \dots & \vdots & \vdots & \vdots & \vdots & \cdots & \vdots \\ 0 & 0 & 0 & \dots & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}, \tag{1.55}$$

$$\hat{H} = \begin{bmatrix} \mathcal{I}_{\mathrm{N-K}} & \mathbf{P}^t \\ \mathbf{0} & \mathbf{0} \end{bmatrix}, \tag{1.56}$$

para assim poder obter uma matriz **G** sistemática como foi visto na equação (1.48).

1.5 Modelo de Geração de Fontes Binárias Distribuídas

O gráfico da Figura 1.10 indica o método para a geração de M fontes correlacionadas $\{U^1, U^2, ..., U^{\mathrm{M}}\}$ usado nesta tese. Cada fonte U^m, $m \in \{1, 2, ..., \mathrm{M}\}$ entrega na sua saída um vetor

$\mathbf{U}^m$ de K_m bits. Cada vetor $\mathbf{U}^m$ é gerado mediante a soma XOR do vetor $\mathbf{U}^0$ e $\mathbf{E}^m$, gerados pelas fontes U^0 e E^m respectivamente. A fonte U^0 tem uma probabilidade $P(u^0 = 1) = p_0$ e as fontes E^m tem uma probabilidade $P(e^m = 1) = p_m$.

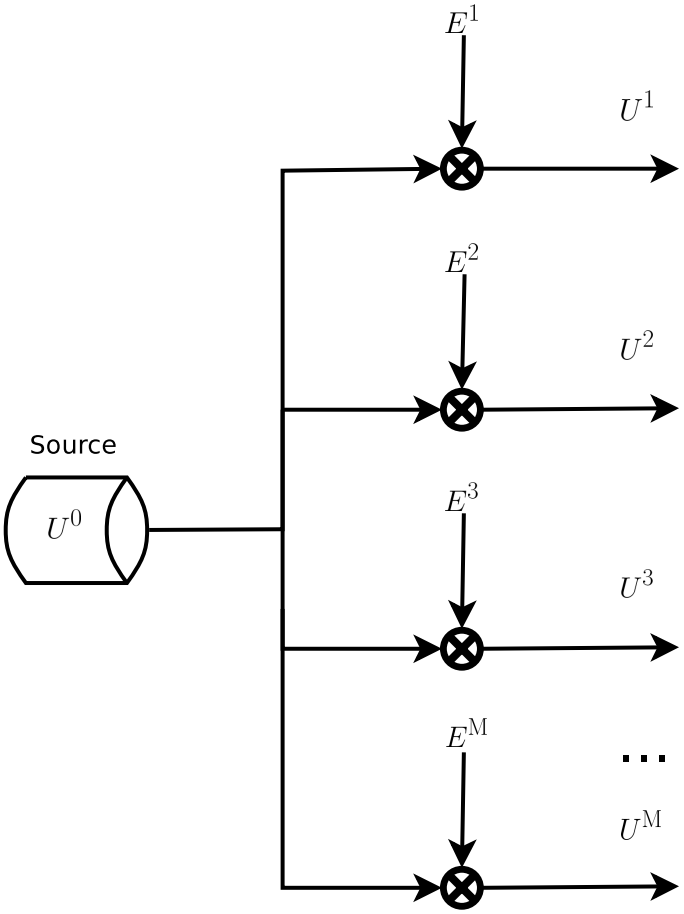

Figura 1.10: Modelo para a geração de dados correlacionados.

Como todas as fontes U^m tem em comum na sua geração a fonte U^0, todas as fontes U^m serão correlacionadas. Que as fontes E^m sejam estatisticamente independentes garante que todas a fontes U^m tenham uma porção de informação que é exclusiva delas.

1.5.1 Correlação no Modelo de Geração de Fontes U^i e U^j com um só Canal BSC

Além do método de geração de fontes correlacionadas usado nesta tese, existem outros métodos. Um destes é o descrito em (Sartipi & Fekri 2008), onde pode-se ver um modelo de fontes correlacionadas definida por

$$P(U^i \neq U^j | U^i) = p_{ij}. \tag{1.57}$$

Isto é equivalente a dizer que se tem um canal BSC de entrada U^i e saída U^j onde se cumprem as seguintes relações

$$\begin{aligned} p_{ij} = P(U^i \neq U^j | U^i) &= P(u^j = 0 | u^i = 1) \\ &= P(u^j = 1 | u^i = 0) \end{aligned}. \tag{1.58}$$

Do anterior deduz-se que U^j é criado passando U^i através de um canal BSC de probabilidade de erro p_{ij}, ver Figura 1.11 (b). No modelo usado em (Sartipi & Fekri 2008), $p_{ij} = p_{ji}$. Isto acontece só num canal BSC com $P(u^i = 1) = P(u^j = 1) = 1/2$, como pode ser visto nas seguintes relações

$$\begin{array}{rcccl} P(u^j = 0, u^i = 1) & = & P(u^j = 0|u^i = 1)P(u^i = 1) & = & p_{ij}P(u^i = 1) \\ & = & P(u^i = 1|u^j = 0)P(u^j = 0) & = & p_{ji}P(u^j = 0)' \end{array} \tag{1.59}$$

$$p_{ji} = p_{ij}\frac{P(u^i = 1)}{1 - P(u^j = 1)}. \tag{1.60}$$

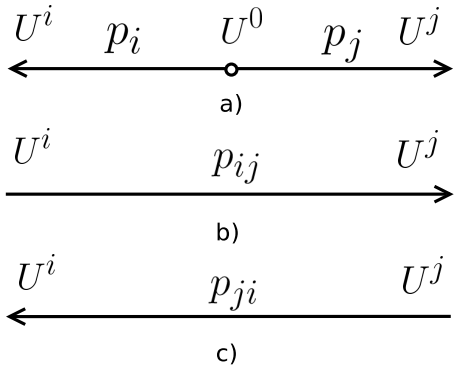

Figura 1.11: a) Duas fontes U^i e U^j geradas enviando U^0 através de dois canais BSC com probabilidade de erro p_i e p_j. b) Duas fontes U^i e U^j, U^j gerada enviando U^i através de um canal BSC de probabilidade de erro p_{ij}. c) Duas fontes U^i e U^j, U^i gerada enviando U^j através de um canal BSC de probabilidade de erro p_{ji}.

Sabendo que no modelo encontrado em (Sartipi & Fekri 2008), U^j é gerado enviando U^i por um canal BSC de probabilidade p_{ij}, e usando o resultado obtido no Apêndice A.7 obtemos que a correlação para este caso é dada por

$$cor(U^i, U^j) = \frac{1 - \frac{p_{ij}}{1-P(u^j=1)}}{\sqrt{\frac{1-P(u^i=1)}{P(u^i=1)}}\sqrt{\frac{P(u^j=1)}{1-P(u^j=1)}}}. \tag{1.61}$$

1.5.2 Correlação no Modelo de Geração de Fontes U^i e U^j com Dois Canais BSC

Nesta seção, para fins didáticos se representará os canais BSC da Figura 1.12 (a) e (b) com um desenho como na Figura 1.12 (c).

O modelo de fontes correlacionadas explicado na Seção 1.5 pode ser representado como uma só fonte U^0 que passa através de M canais BSC, gerando assim M fontes correlacionadas U^m, $\forall m \in \{1, 2, ..., \mathrm{M}\}$. Na Figura 1.13 pode-se ver um exemplo de cinco fontes correlacionadas na sua representação com canais BSC. Cada canal terá sua própria probabilidade de erro p_m. Esta probabilidade de erro p_m é a mesma probabilidade $P(e^m = 1) = p_m$ da fonte E^m do modelo de geração de fontes da Figura 1.10.

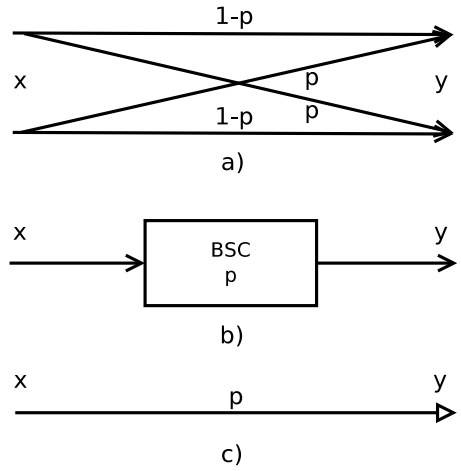

Figura 1.12: a), b) Canal BSC de probabilidade de erro p. c) Representação equivalente de um canal BSC de probabilidade de erro p.

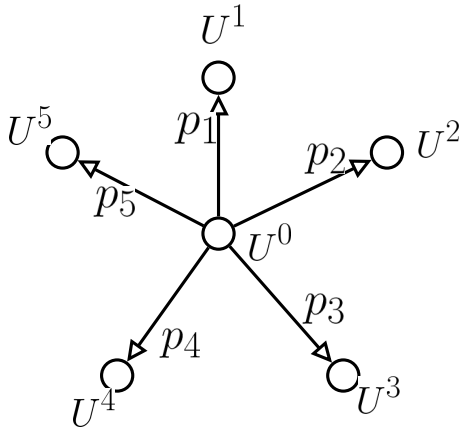

Figura 1.13: Cinco fontes correlacionadas, modelo usando canais BSC de probabilidade p_m.

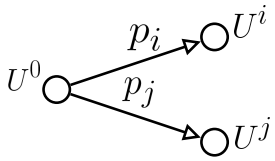

Figura 1.14: Duas fontes U^i e U^j geradas a partir de uma fonte U^0.

Dado que cada ramo da Figura 1.13 é independente, toma-se para análise dois ramos quaisquer como ilustrado na Figura 1.14.

Define-se p_{u^i} como a probabilidade $P(U^i = 1)$. Sendo assim para todo $i \neq j/\ i, j \in \{1, 2, ..., \mathrm{M}\}$ se sabe que

$$p_{u^i} = p_i + p_0 - 2p_i p_0, \tag{1.62}$$

$$p_{u^j} = p_j + p_0 - 2p_j p_0. \tag{1.63}$$

A correlação entre as fontes U^i e U^j esta dada pela equação (1.64) como pode ser visto no

Apêndice A.3

$$corr(U^i, U^j) = \frac{1 - 2(p_i + p_j - 2p_i p_j)}{\sqrt{1 + \frac{p_i(1-p_i)(1-2p_0)}{p_0(1-p_0)}}\sqrt{1 + \frac{p_j(1-p_j)(1-2p_0)}{p_0(1-p_0)}}}. \tag{1.64}$$

Analisando o caso específico em que $P(u^0 = 1) = p_0 = 0.5$ obtemos

$$corr(U^i, U^j) = 1 - 2\{p_i + p_j - 2p_i p_j\} \tag{1.65}$$

ou

$$corr(U^i, U^j) = (1 - 2p_i)(1 - 2p_j). \tag{1.66}$$

Neste trabalho, quando p_i for igual a p_j se usará a letra ρ para referir-se a ambos.

Capítulo 2

Codificação Conjunta Fonte-Canal para Fontes Distribuídas

Neste capítulo três contribuições serão apresentadas. As duas primeiras são a obtenção de duas formas, para dois casos distintos, do cálculo da taxa de codificação fonte-canal ótima em termos de uso de energia do codificador. Isto será visto na Seção 2.3. A terceira contribuição será a definição de uma variante do limite de Shannon para a capacidade do canal em fontes distribuídas, isto será tratado na Seção 2.4.

2.1 Introdução

Para a codificação fonte-canal se parte da ideia que há M fontes correlacionadas $U^m, m \in \{1, 2, ..., \mathrm{M}\}$. Estas fontes são codificadas independentemente para diminuir a redundância delas e entre elas (codificação independente de fontes distribuídas), e também para proteger sua informação quando ela percorre os M canais $BI - AWGN$ (do inglês "Binary Input - Additive White Gaussian Noise"). Uma codificação de fontes não distribuídas baseia-se fundamentalmente em tirar a informação redundante, entendendo como redundante qualquer informação a mais de $H(U^m)$, por outro lado a codificação de fontes distribuídas visa diminuir a informação redundante seguindo o teorema $Slepian - Wolf$, entendendo como redundante toda informação maior que R_m. Neste tese a decodificação se faz de forma conjunta como é ilustrado na Figura 2.1.

Cada uma das M fontes tem $H(U^m)$ de informação. Seguindo o teorema $Slepian - Wolf$ se sabe que existe um conjunto de taxas de informação R_m correspondentes a cada fonte U^m, de modo que o conhecimento da informação R_m de todos os U^m possa ser suficiente para recuperar toda informação das fontes U^m. Isto implica que todas as fontes tem como redundância $H(U^m) - \mathrm{R}_m$ de informação. Este excedente de informação pode ser aproveitado no momento da codificação do canal, porque cada fonte já possui redundância. Sabendo disso, é preciso só acrescentar uma quantidade menor de redundância na codificação de canal de modo que se tenha uma taxa de codificação fonte-canal r_m ótima para o canal U^m. Isto não quer dizer que R_m não tenha já informação redundante com as outras fontes, senão que esta taxa de informação garante que obtendo a informação R_m de todos os canais, a informação das fontes U^m é recuperável.

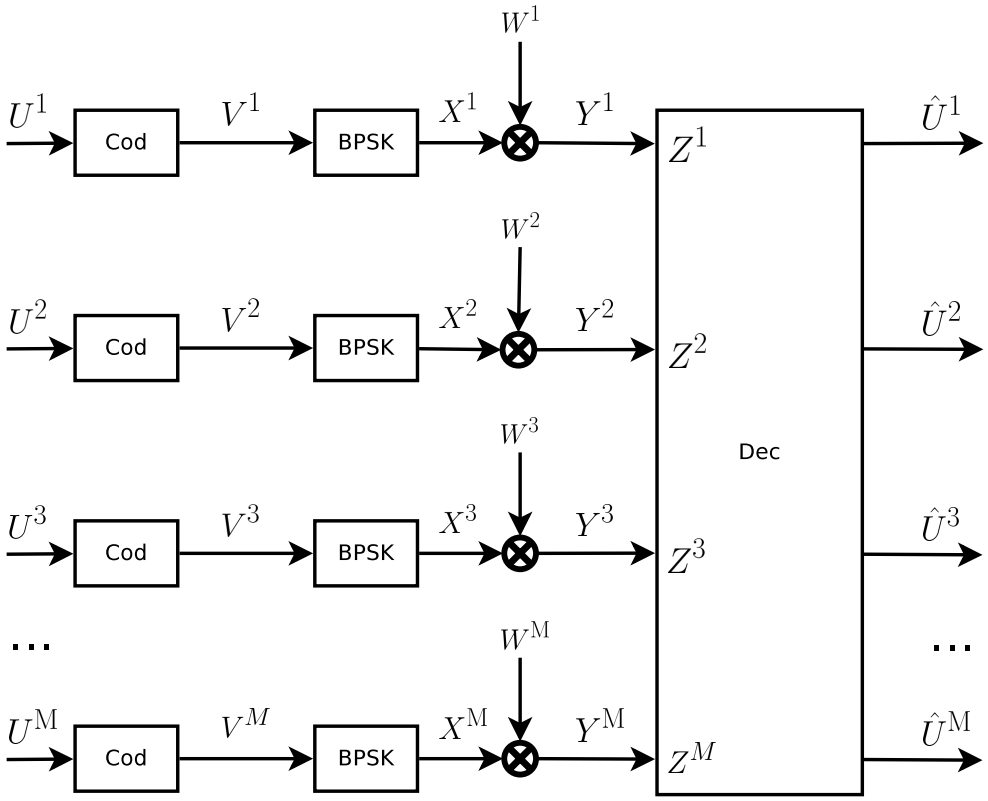

Figura 2.1: Codificador fonte-canal e decodificador conjunto.

Cada fonte de informação tem um codificador fonte-canal de taxa de codificação $\mathrm{r}_m = \mathrm{K}_m/\mathrm{N}_m$, onde K_m é o tamanho da palavra de $\mathbf{U}^m$ e N_m é o tamanho da palavra codificada $\mathbf{V}^m$. Se usará $H(U^m)$ para definir a informação de cada bit que sai da fonte. Todas as palavras codificadas $\mathbf{V}^m$ serão moduladas usando modulação binária por deslocamento de fase $BPSK$ (do inglês "Binary Phase Shift Keying"), para logo serem enviadas por canais $BI-AWGN$, seguindo a equação

$$Y^m = X^m + W^m, \quad \forall m \in \{1, 2, ..., \mathrm{M}\}. \tag{2.1}$$

X^m é a informação na saída do demodulador, Y^m é a saída do canal $BI-AWGN$ e W^m é o ruído $AWGN$ no canal $m-ésimo$, com $E[W^m] = 0$ e $E[(W^m)^2] = \sigma_m^2$. Para finalizar, a decodificação será feita de forma conjunta de modo que $(\hat{U}_1, \hat{U}_2, ..., \hat{U}_{\mathrm{M}})$ seja obtido a partir de $(Y^1, Y^2, ..., Y^{\mathrm{M}})$

$$(\hat{U}^1, \hat{U}^2, ..., \hat{U}^{\mathrm{M}}) = f(Y^1, Y^2, ..., Y^{\mathrm{M}}). \tag{2.2}$$

Também é possível usar $(Z^1, Z^2, ..., Z^{\mathrm{M}})$ que é a decisão abrupta de $(Y^1, Y^2, ..., Y^{\mathrm{M}})$. Obtendo assim a regra de decisão

$$(\hat{U}^1, \hat{U}^2, ..., \hat{U}^{\mathrm{M}}) = f(Z^1, Z^2, ..., Z^{\mathrm{M}}). \tag{2.3}$$

2.2 Codificação de Fonte e de Canal

2.2.1 Codificação de Fonte

Entre os métodos de codificação de fonte temos o estudado em (Pradhan & Ramchandran 2003). Este método usa síndromes para codificar o vetor de informação U^m, $\forall m \in \{1, ..., \mathrm{M}\}$. Seguindo esta linha se tem o trabalho de (Schonberg, Ramchandran & Pradhan 2004), onde se codifica usando síndromes e a decodificação é feita usando grafo fatores (Kschischang, Frey & Loeliger 2001) junto com o algoritmo soma-produto (SP). Na Figura 2.2 (a) pode-se ver um grafo fator que representa a codificação de fonte (independente) dos vetores $\mathbf{U}^i = \{u_0^i, u_1^i, ..., u_{\mathrm{K}_i-1}^i\}$ e $\mathbf{U}^j = \{u_0^j, u_1^j, ..., u_{\mathrm{K}_j-1}^j\}$ usando síndromes e matrizes de tipo $LDPC$, gerando os vetores $\mathbf{Q}^i = \{q_0^i, q_1^i, ..., q_{l_i-1}^i\}$ e $\mathbf{Q}^j = \{q_0^j, q_1^j, ..., q_{l_j-1}^j\}$, respectivamente. Um ponto importante a ressaltar nesta imagem é que o vetor $\mathbf{U}^j$ é gerado pela passagem do vetor $\mathbf{U}^i$ através de um canal BSC. Este foi o modelo de geração de fontes adotado em (Schonberg et al. 2004). Na Figura 2.2 (b) pode-se ver uma representação compacta das duas fontes.

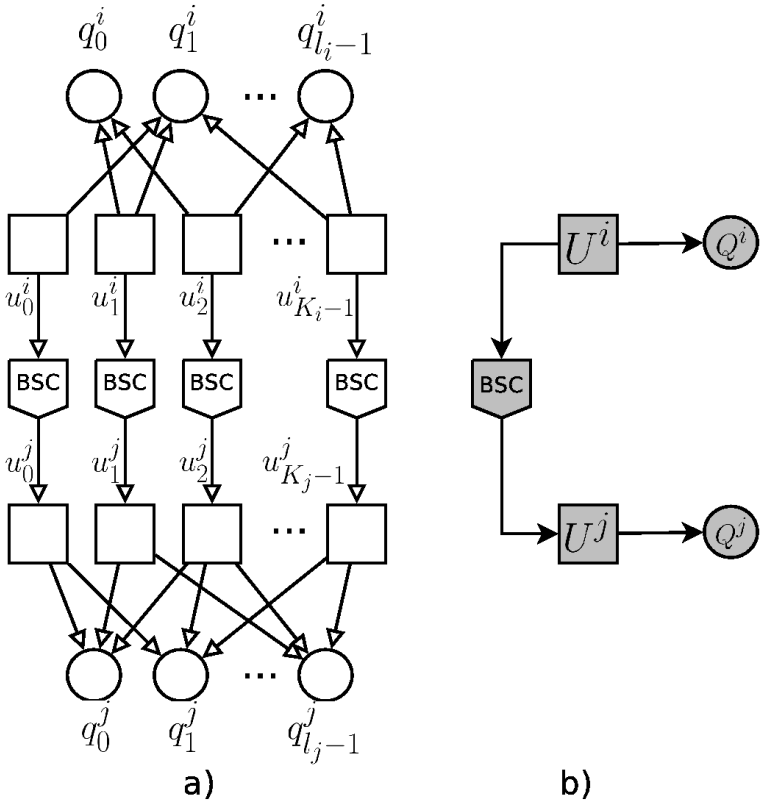

Figura 2.2: (a) Representação mediante grafo fator do modelo de geração e codificação de fontes. (b) Representação compacta do grafo fator desenhado em (a).

Na Figura 2.3 (a) pode-se ver uma representação mediante grafo fator de M vetores de informação $\mathbf{U}^m$, $\forall m \in \{1, 2, ..., \mathrm{M}\}$, geradas pelo modelo de fonte em (Schonberg et al. 2004). Os vetores $\mathbf{Q}_m$ de l_m bits, representam a informação comprimida mediante síndromes e cumprem o teorema $Slepian - Wolf$. A Figura 2.3 (b) representa mediante grafo fator o mesmo método de codificação com a diferença que se aplica o modelo de geração de fontes explicado na Seção

1.5, onde $\mathbf{U}^0$ é um vetor fonte que passa por M canais BSC com distintas probabilidades de erro de bit para gerar as fontes U^m. A Figura 2.3 (b) será o modelo de geração de fontes usado nesta tese.

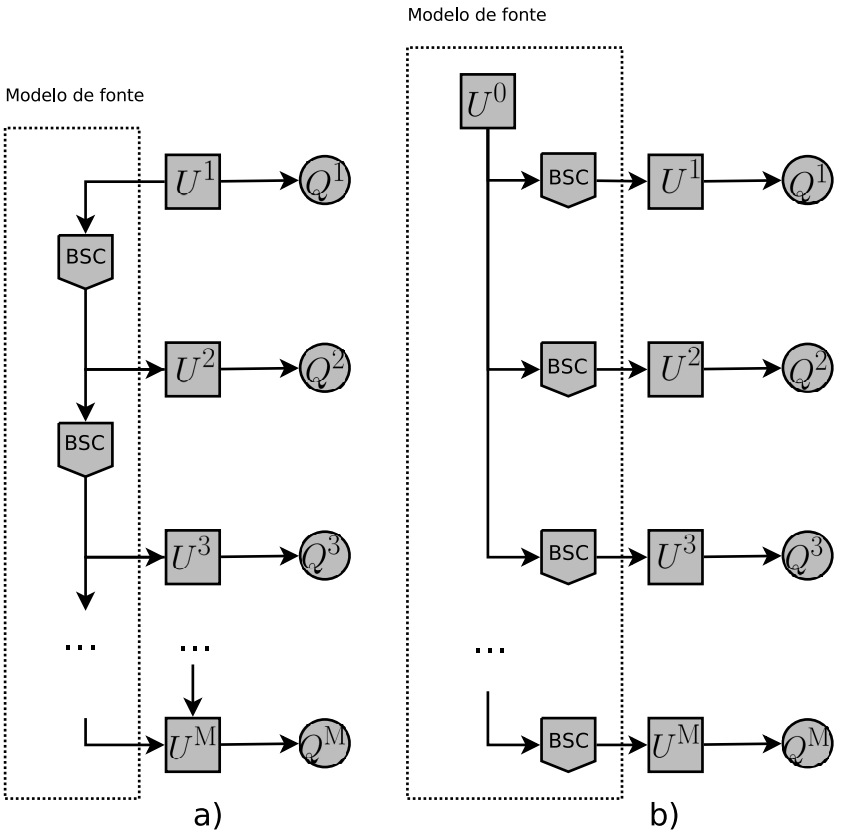

Figura 2.3: (a) Representação mediante grafo fator do modelo de geração de fontes e codificação em modo cascata . (b) Representação mediante grafo fator do modelo de geração de fontes usado nesta tese.

2.2.2 Codificação de Canal

Temos na saída do codificador de fonte um vetor chamado $\mathbf{Q}^m$ de l_m bits. Este vetor de informação é codificado para proteger o vetor quando da sua passagem por um canal com capacidade de canal C_m. Na Figura 2.4 tem-se um diagrama que mostra o antes explicado, onde pode-se ver que na entrada do codificador, tem-se um vetor $\mathbf{Q}^m$ com uma informação igual

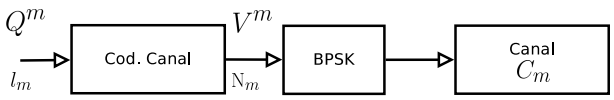

Figura 2.4: Codificação de canal com um canal de capacidade de canal C_m.

a $H(\mathbf{Q}^m)$. Conhecendo isto, pode-se dizer que a taxa de informação na saída do codificador de canal é

$$\mathrm{r}_m^c = \frac{H(\mathbf{Q}_m)}{\mathrm{N}_m}. \tag{2.4}$$

Por outro lado, do teorema de Shannon sabe-se que

$$\mathrm{r}_m^c \leq \mathrm{C}_m. \tag{2.5}$$

Então, para proteger o vetor $\mathbf{Q}^m$ no percurso no canal, é necessário que

$$\frac{H(\mathbf{Q}^m)}{\mathrm{N}_m} \leq \mathrm{C}_m. \tag{2.6}$$

2.2.3 Codificação Não Conjunta

Consideram-se dois tipos de codificação, uma codificação de fonte e uma codificação de canal. A codificação de fonte usada nesta tese é uma codificação independente de fontes distribuídas. Esta visa atingir as taxas de informação especificadas pelo teorema de $Slepian - Wolf$. Para isto se tomam K_m bits de cada uma das fontes U^m para obter um vetor de entrada $\mathbf{U}^m$. Este se codifica com l_m bits gerando o vetor $\mathbf{Q}^m$, onde $\mathrm{K}_m \geq l_m$. Sendo assim, a redundância tirada pela codificação é $H(\mathbf{U}^m) - H(\mathbf{Q}^m)$.

A codificação de canal visa proteger a informação do vetor $\mathbf{Q}^m$ ao passar por um canal ruidoso. Para isto acrescenta-se redundância gerando um vetor $\mathbf{V}^m$ de N_m bits, onde $l_m \leq \mathrm{N}_m$.

Na Figura 2.5 pode-se ver como os M vetores $\mathbf{U}^m$, $\forall m \in \{1, ..., \mathrm{M}\}$ recebem uma codificação independente de fontes distribuídas (Codificação $Slepian - Wolf$) e uma codificação de canal. Após esta codificação não se tem garantias de que $\mathrm{K}_m \leq \mathrm{N}_m$.

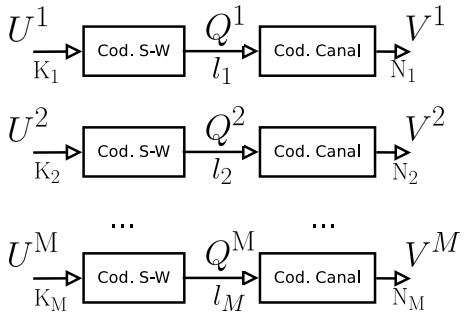

Figura 2.5: Codificação de fonte e codificação de canal.

2.2.4 Codificação Conjunta

A codificação conjunta, também chamada de codificação fonte-canal tem como objetivo usar apenas um bloco codificador para conseguir a codificação de fonte e a codificação de canal. A codificação fonte-canal usada nesta tese é independente, ou seja sem comunicação entre codificadores.

Para chegar de $\mathbf{U}^m$ a $\mathbf{V}^m$ usando um só codificador, tem-se que fazer uma análise distinta para o caso em que $\mathrm{K}_m \geq \mathrm{N}_m$ e uma outra em que $\mathrm{K}_m \leq \mathrm{N}_m$.

Por exemplo se $\mathrm{K}_m \geq \mathrm{N}_m$, quer dizer que depois de tirar a informação comum redundante e acrescentar redundância para percorrer o canal, ainda assim, se obtém um vetor $\mathbf{V}^m$ de comprimento menor que $\mathbf{U}^m$. Por conta disto não seria necessário acrescentar redundância

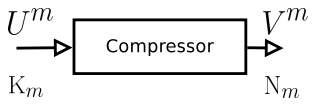

Figura 2.6: Comprimindo a fonte U^m.

(codificar) a $\mathbf{U}^m$. Poderia-se comprimir de modo que se tenha informação redundante suficiente para recuperar o vetor $\mathbf{U}^m$ após o canal, ver Figura 2.6.

Se $\mathrm{K}_m \leq \mathrm{N}_m$, então seria possível codificar a fonte e acrescentar redundância usando só um codificador com uma taxa $\mathrm{r}_m = \mathrm{K}_m/\mathrm{N}_m \leq 1$, ver Figura 2.7.

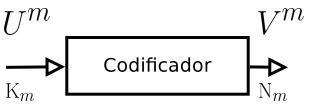

Figura 2.7: Codificando a fonte U^m.

Em geral pode-se trabalhar a codificação de fonte e canal de forma separada ou de forma conjunta.

2.3 Taxas Ótimas Para Codificação Conjunta

Com base nas seções 2.2.1 e 2.2.2, sabe-se que a codificação de fonte e a de canal pode ser feita usando matrizes. Como visto, ambas as matrizes estão em série e, portanto, estas podem ser substituídas por uma só matriz. Assim, teremos uma codificação conjunta.

2.3.1 Codificação Independente de Fonte-Canal de Fontes Distribuídas

Analisando separadamente a codificação de fonte e de canal, sabe-se que o codificador de fonte recebe um vetor de entrada $\mathbf{U}^m$ de K_m elementos, com uma informação total de $H(\mathbf{U}^m)$. Na saída deste codificador (codificador $Slepian - Wolf$) independente para fontes distribuídas tem-se um vetor $\mathbf{Q}^m$ com uma informação de

$$H(\mathbf{Q}^m) = \mathrm{K}_m \mathrm{R}_m. \tag{2.7}$$

O vetor $\mathbf{Q}^m$ é enviado ao codificador de canal gerando um vetor $\mathbf{V}^m$ de N_m elementos, ver Figura 2.8. Estes codificadores em série geram uma taxa de codificação fonte-canal

$$\mathrm{r}_m = \frac{\mathrm{K}_m}{\mathrm{N}_m}. \tag{2.8}$$

De (2.6), (2.7) e (2.8) obtemos

$$\mathrm{R}_m \leq \frac{\mathrm{C}_m}{\mathrm{r}_m}. \tag{2.9}$$

Na Figura 2.9 pode-se ver uma representação visual da equação (2.9). Misturando esta equação com a equação (1.16) do teorema de $Slepian - Wolf$, se obtém

$$H(U(S)|U(S^c)) - \sum_{m \in S} \frac{\mathrm{C}_m}{\mathrm{r}_m} \leq 0. \tag{2.10}$$

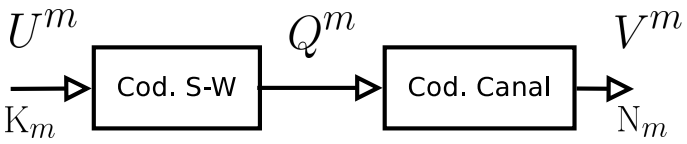

Figura 2.8: Codificador de fonte e de canal para a fonte U^m.

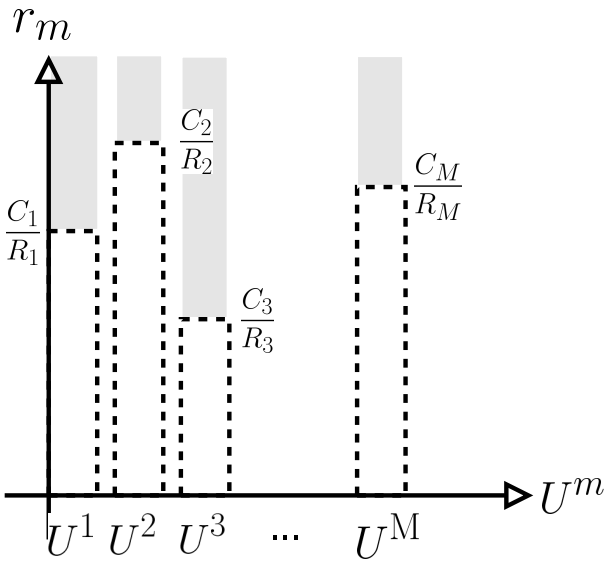

Figura 2.9: Gráficos das taxas de codificação fonte-canal das M fontes, seguindo a equação (2.9).

Em (Garcia-Frias, Zhao & Zhong 2007), uma região factível para taxas R_m foi representada fixando C_m e r_m, enquanto que em (Abrardo, Ferrari, Martalò, Franceschini & Raheli 2012), uma região factível para C_m foi descrita fixando r_m. Já em (Yedla, Pfister & Narayanan 2013) somente as taxas r_m foram fixadas. Como sempre, as capacidades C_m são dependentes dos parâmetros do canal e uma região factível é descrita em termos de outros parâmetros. Nas seguintes seções se estuda o caso onde é fixado C_m e define-se dois diferentes problemas de otimização para as taxas r_m. Em ambos problemas a região factível é dada por (2.10).

2.3.2 Máxima Taxa Comum

Neste caso está-se assumido que todas as taxas de codificação de fonte-canal r_m são iguais a r e que a variável a ser maximizada é a taxa comum r. Usando (2.10) o problema de otimização é definido como segue

$$r_{max} = \max\{r\} \tag{2.11}$$

sujeito a restrição

$$r \leq \frac{\sum_{m\in S} \mathrm{C}_m}{H(U(S)|U(S^c))}. \tag{2.12}$$

O máximo valor de r é dado pelo valor mínimo do lado direito da equação (2.12),

$$r_{max} = \min_S \{\frac{\sum_{m\in S} \mathrm{C}_m}{H(U(S)|U(S^c))}\}. \tag{2.13}$$

Exemplo 2.3.1 (Duas fontes correlacionadas com taxa comum) *Sendo U^1 e U^2 duas fontes binárias correlacionadas com $H(U^1|U^2) = H(U^2|U^1) = h(p)$ e $H(U^1U^2) = 1 + h(p)$, onde $h(p)$ é a função da entropia binária. As capacidades dos dois canais ortogonais estão relacionadas por $\mathrm{C}_1 = \mathrm{C}_2/2 = \mathrm{C}$. A taxa ótima r será então dada pelo máximo valor de*

$$\frac{\mathrm{C}}{h(p)}, \frac{2C}{h(p)}, \frac{3C}{1+h(p)}. \tag{2.14}$$

Assumindo que $h(p) > 0.5$ temos que taxa ótima

$$r_{max} = \frac{\mathrm{C}}{h(p)}. \tag{2.15}$$

Exemplo 2.3.2 (M fontes correlacionadas com taxa comum) *Considerando agora o caso onde todas as capacidades de canal são iguais a $\mathrm{C}_m = \mathrm{C}$, $m \in \{1, 2, .., \mathrm{M}\}$, e além disso se tem um modelo de correlação onde todas as variáveis U^m, $m \in \{1, 2, .., \mathrm{M}\}$, são a saída de um conjunto de canais BSC com igual probabilidade de erro. A entrada destes canais é a uma fonte binária comum de valores equiprováveis . Este modelo foi assumido, por exemplo, em (K. Kobayashi & Katayama 2009), (Del Ser, Garcia-Frias & Crespo 2009) e (Abrardo et al. 2012). A entropia conjunta relacionada a este modelo é uma função do número de fontes envolvidas e a probabilidade de erro comum. As restrições em (2.12) são agora dadas por*

$$r \leq \frac{|S|\mathrm{C}}{H(U(S)|U(S^c))}, \tag{2.16}$$

onde $|S|$ é a cardinalidade do conjunto S. É provado no Apêndice A.15 que

$$r_{max} = \frac{\mathrm{MC}}{H(U^1U^2...U^{\mathrm{M}})}. \tag{2.17}$$

Vale a pena notar que o limite teórico calculado em (K. Kobayashi & Katayama 2009) e (Del Ser et al. 2009) pode ser obtido usando (2.17). Além disso é importante notar que a capacidade balanceada definida em (Abrardo et al. 2012) pode ser derivado de (2.17).

2.3.3 Máxima Taxa Soma

Agora considera-se o problema de maximizar a taxa soma $\sum_{m=1}^{\mathrm{M}} \mathrm{r}_m$, sob a mesma restrição descrita por (2.10). Este problema não pode ser transformado num problema convexo. Embora isso possa ser solucionado numericamente, aqui se define o seguinte problema equivalente. Assumindo que os valores de $\mathrm{K}_m = \mathrm{K}$ para todas as fontes U^m, um problema equivalente é minimizar a soma $\sum_{m=1}^{\mathrm{M}} \mathrm{N}_m$ sujeito a

$$H(U(S)|U(S^c))\mathrm{K} \leq \sum_{m \in S} \mathrm{N}_m \mathrm{C}_m. \tag{2.18}$$

Este problema equivalente é descrito, a partir da equação (2.19), em forma de programação linear inteira ou ILP (do inglês "Integer Linear Programming"). Seja $\mathbf{N} = (\mathrm{N}_1, \mathrm{N}_2, ..., \mathrm{N}_\mathrm{M})^T$, onde o índice superior T denota transposição. O problema ILP pode agora ser descrito como

$$\min_{\mathbf{N}} \mathbf{F}^T\mathbf{N} \tag{2.19}$$

sujeito a

$$\mathbf{AN} \leq \mathbf{B}, \tag{2.20}$$

onde

$$\mathbf{F} = (1, 1, ..., 1)^T, \tag{2.21}$$

$$\mathbf{B} = \mathrm{K}\begin{pmatrix} H(U(S_1)|U(S_1^c)) \\ H(U(S_2)|U(S_2^c)) \\ \vdots \\ H(U(S_{2^\mathrm{M}-1})|U(S_{2^\mathrm{M}-1}^c)) \end{pmatrix}, \tag{2.22}$$

$$\mathbf{A} = \mathbf{DC}, \tag{2.23}$$

com

$$\mathbf{D} = \begin{pmatrix} 0 & 0 & \cdots & 0 & 0 & 1 \\ 0 & 0 & \cdots & 0 & 1 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & 1 & \cdots & 1 & 1 & 1 \end{pmatrix} \tag{2.24}$$

e

$$\mathbf{C} = diag(\mathrm{C}_1, \mathrm{C}_2, ..., \mathrm{C}_\mathrm{M}) \tag{2.25}$$

é uma matriz diagonal. A matriz coluna $\mathbf{F}$ tem M elementos, a matriz coluna $\mathbf{B}$ tem $2^\mathrm{M} - 1$ elementos e a matriz $\mathbf{D}$ tem $2^\mathrm{M} - 1$ linhas e M colunas. As linhas de $\mathbf{D}$ são representações binárias de inteiros $i \in \{1, 2, ..., 2^\mathrm{M} - 1\}$. Portanto, os uns em cada linha indicam que fontes pertencem ao subconjunto S_i.

Exemplo 2.3.3 (15 fontes correlacionadas com taxa variável) *Considerando agora* $\mathrm{M} = 15$ *e* $\mathrm{K} = 1000$. *As fontes tem a mesma correlação que no modelo do Exemplo 2.3.2 onde a probabilidade de erro dos canais BSC é igual a* 0.7 *e a fonte comum emite bits de forma equiprovável. As capacidades* C_m, $m \in \{1, \ldots, \mathrm{M}\}$, *estão dadas na Tabela 2.1 e foram geradas aleatoriamente com valores entre* 0.2 *e* 0.8. *Aplicando o algoritmo "branch-and-bound " (junto com cortes inteiros de Gomory) (GPLK 2000) ao problema ILP da equação (2.19), obtém-se a correspondentes comprimentos ótimos mostrados na Tabela 2.1.*

Exemplo 2.3.4 (15 fontes correlacionadas com taxa variável e $\mathrm{K}_m = 100000$**)** *Usando os mesmos dados e o algoritmo do Exemplo 2.3.3 (GPLK 2000), com a diferença de que se tem valores de* $\mathrm{K}_m = 100000$, *obtêm-se valores de* N_m *como mostra a Tabela 2.2. Na resolução do algoritmo pode-se observar que este não apresenta um maior aumento de complexidade pelo incremento do valor de* K_m, *em geral pode-se dizer que o algoritmo cresce em complexidade só com o aumento do número de fontes* M, *sendo este crescimento exponencial.*

Tabela 2.1: Capacidades C_m e correspondente longitude ótima N_m para $\mathrm{K}_m = 1000$

C_m	N_m	C_m	N_m	C_m	N_m
7.55884051e-01	1322	7.47679003e-01	1313	7.45078468e-01	1298
7.23930249e-01	1322	6.73303774e-01	1407	5.75244526e-01	1633
5.55729071e-01	1679	5.45227625e-01	1700	4.00368675e-01	2300
3.95929714e-01	2319	3.66826321e-01	2490	3.39960378e-01	2677
2.55462086e-01	3550	2.50141329e-01	3616	2.17808960e-01	4142

Tabela 2.2: Capacidades C_m e correspondente longitude ótima N_m para $\mathrm{K}_m = 100000$

C_m	N_m	C_m	N_m	C_m	N_m
7.55884051e-01	132295	7.47679003e-01	131150	7.45078468e-01	130002
7.23930249e-01	132131	6.73303774e-01	140688	5.75244526e-01	163299
5.55729071e-01	167821	5.45227625e-01	169993	4.00368675e-01	230214
3.95929714e-01	231710	3.66826321e-01	249019	3.39960378e-01	267693
2.55462086e-01	355055	2.50141329e-01	361539	2.17808960e-01	414118

Exemplo 2.3.5 (M fontes correlacionadas com taxa variável) *Considere agora que as fontes tem o mesmo modelo de correlação do Exemplo 2.3.3, mas com* $\mathrm{C}_m = \mathrm{C}, \forall m \in \{1, \ldots, \mathrm{M}\}$. *As restrições de (2.18) são reescritas como*

$$\sum_{m \in S} \mathrm{N}_m \geq \frac{\mathrm{K}}{\mathrm{C}} H(U(S)|U(S^c)). \tag{2.26}$$

Assumindo-se que $\mathbf{N} = (\mathrm{N}_1, \mathrm{N}_2, \ldots, \mathrm{N}_\mathrm{M})^T$ *é um vetor com valores reais. O hiperplano definido pela solução de*

$$\sum_{m=1}^{M} \mathrm{N}_m = \frac{\mathrm{K}}{\mathrm{C}} H(U_1, U_2, ..., U_M), \tag{2.27}$$

representa o conjunto de soluções que minimiza a soma $\sum_{m=1}^{\mathrm{M}} \mathrm{N}_m$. *Uma possível solução é obtida fixando* $\mathrm{N}_m = \mathrm{N}$ *para todo valor de* m, *i.e.,* $\mathrm{N}_{min} = \frac{\mathrm{K}}{\mathrm{MC}} H(U_1, U_2, ..., U_M)$. *Como sempre,* $r = \mathrm{K}/\mathrm{N}_{min}$ *é a mesma taxa dada por (*2.13*). Portanto pode ser concluído que:*

Proposição 2.1 *Para canais ortogonais com capacidades iguais e o modelo de correlação do Exemplo 2.3.2, a máxima taxa soma é obtida fixando a codificação de cada fonte com uma única taxa comum.*

2.4 Limite de Shannon

O limite de Shannon para um canal de comunicações é a máxima taxa teórica de informação que pode ser transmitida no canal dado um nível de ruído. Na equação (2.9) foi visto que para

ter uma comunicação livre de ruído é necessário que a multiplicação das taxas $\mathrm{r}_m\mathrm{R}_m$, seja menor que a capacidade do canal C_m,

$$\mathrm{r}_m\mathrm{R}_m \leq \mathrm{C}_m. \tag{2.28}$$

Em outras palavras, a quantidade de informação codificada $\mathrm{r}_m\mathrm{R}_m$ tem que ser menor que a quantidade máxima de informação C_m que o canal pode transportar de forma confiável. Note que a capacidade do canal C_m varia em função da razão sinal-ruído E_b/N_0, onde E_b é a energia para transmitir um bit de informação e $N_0/2 = \sigma^2$ é a densidade espectral de potência do ruído de banda-dupla no canal $AWGN$. O limite de Shannon é atingido quando a equação (2.28) está no limite de ser insatisfeita, ou seja

$$\mathrm{r}_m\mathrm{R}_m = \mathrm{C}_m. \tag{2.29}$$

A Figura 2.10 representa a melhor curva do desempenho de um código corretor de erro. Para valores da razão sinal-ruído menores que o limite de Shannon é impossível obter uma taxa de erro de bit, BER (do inglês "Bit Error Rate"), igual a zero na estimação de U^m.

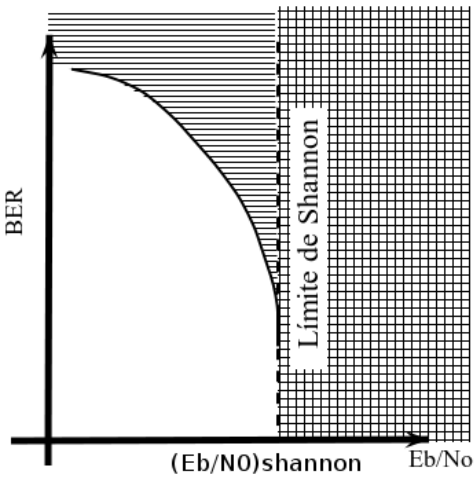

Figura 2.10: Gráfico de BER versus E_b/N_0 atingindo o limite de Shannon.

2.4.1 Limite de Shannon para Fontes Não Distribuídas

No caso de fontes não distribuídas a equação (2.29) é rescrita para $\mathrm{R}_m = 1$, obtendo-se

$$\mathrm{r}_m = \mathrm{C}_m. \tag{2.30}$$

Nas equações (1.33) e (1.38) se descrevem os valores de capacidade de canal em função de E_b/N_0 para os canais do tipo BI-$AWGN$ e *hard-decision* BI-$AWGN$. Usando estas equações junto com a equação (2.30) calcula-se o limite de *Shannon* para um canal BI-$AWGN$ como

$$(E_b/N_0)_{BI-AWGN} = \frac{1}{2\mathrm{r}_m\phi^{-1}(\mathrm{r}_m)} \tag{2.31}$$

e o limite de *Shannon* para um canal *hard-decision* BI-$AWGN$ como

$$(E_b/N_0)_{HD-BI-AWGN} = \frac{\{Q^{-1}(h^{-1}(1-\mathrm{r}_m))\}^2}{2\mathrm{r}_m}. \tag{2.32}$$

Sendo $\phi^{-1}(.)$, $Q^{-1}(.)$ e $h^{-1}(.)$ funções inversas de $\phi(.)$, $Q(.)$ e $h(.)$ respectivamente.

A Tabela 2.3 compara os limites de Shannon para um canal $BI\text{-}AWGN$ e um canal $BI\text{-}AWGN$ com decisão abrupta. Os valores limites do E_b/N_0 para valores de taxa de código tendendo a zero foram calculados nos Apêndices A.19 e A.20.

Rc	$(E_b/N_0)_{BI-AWGN}$ (dB)	$(E_b/N_0)_{HD-BI-AWGN}$ (dB)
0.00000	-1.591745	0.369453
0.05000	-1.440140	0.479204
0.10000	-1.285578	0.593859
0.15000	-1.127028	0.713921
0.20000	-0.963520	0.839975
0.25000	-0.794059	0.972710
0.30000	-0.617563	1.112949
0.35000	-0.432803	1.261682
0.40000	-0.238341	1.420119
0.45000	-0.032437	1.589760
0.50000	0.187060	1.772501
0.55000	0.422898	1.970789
0.60000	0.678679	2.187866
0.65000	0.959306	2.428168
0.70000	1.271732	2.698015
0.75000	1.626371	3.006897
0.80000	2.039998	3.370134
0.85000	2.542627	3.815174
0.90000	3.197745	4.400136
0.95000	4.191145	5.295271

Tabela 2.3: Limite de Shannon para os canais $BI\text{-}AWGN$ e hard-decision $BI\text{-}AWGN$.

2.4.2 Limite Virtual de Shannon para Fontes Distribuídas

De forma semelhante que na Seção 2.4.1, no caso de fontes distribuídas usa-se a equação (2.29) junto com a equação (1.28) para obter no caso de um canal do tipo $BI\text{-}AWGN$ a seguinte relação

$$(E_s/N_0)_{BI-AWGN} = \frac{1}{2\phi^{-1}(\mathrm{r}_m\mathrm{R}_m)}. \tag{2.33}$$

Nesse sentido usando a equação (1.37) se obtém para um canal *hard-decision* $BI\text{-}AWGN$

$$(E_s/N_0)_{HD-BI-AWGN} = \frac{\{Q^{-1}(h^{-1}(1-\mathrm{r}_m\mathrm{R}_m))\}^2}{2}. \tag{2.34}$$

Para obter o limite de Shannon é necessário achar uma relação entre E_b e E_s para uma codificação fonte-canal. No gráfico da Figura 2.11 pode-se ver E_b e $\hat{E}_b$, sendo estes valores a energia que gasta cada bit nessa posição para ser transmitido. Deste modo se deduz que

$$E_b = \frac{E_s}{\mathrm{r}_m}, \tag{2.35}$$

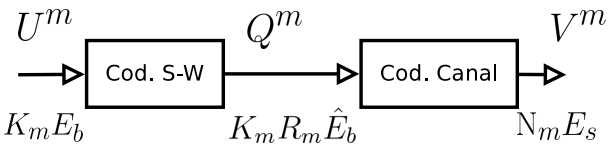

Figura 2.11: Diagrama de blocos mostrando a energia $\hat{E}_b$ do bit a ser transmitido.

$$\hat{E}_b = \frac{E_s}{\mathrm{r}_m \mathrm{R}_m}. \tag{2.36}$$

E_b é a energia que se gasta no envio de cada bit com informação redundante (seguindo o teorema de $Slepian - Wolf$). $\hat{E}_b$ é a energia que se gasta no envio de cada bit sem informação redundante, isto é, a energia que gasta cada bit com a mínima expressão da informação de U_m. Assim usando as equações (2.33), (2.34) e (2.36) se obtém o limite virtual de Shannon para um canal $BI - AWGN$ como

$$(\hat{E}_b/N_0)_{BI-AWGN} = \frac{1}{2\mathrm{r}_m \ \mathrm{R}_m \phi^{-1}(\mathrm{r}_m \mathrm{R}_m)} \tag{2.37}$$

e para *hard-decision BI-AWGN* como

$$(\hat{E}_b/N_0)_{HD-BI-AWGN} = \frac{\{Q^{-1}(h^{-1}(1 - \mathrm{r}_m \mathrm{R}_m))\}^2}{2\mathrm{r}_m \mathrm{R}_m}. \tag{2.38}$$

Exemplo 2.4.1 (Limite virtual de Shannon para fontes com correlações iguais) *Considere o caso específico do Exemplo 2.3.2 onde todas as capacidades de canal* C_m *são iguais a* C*, as taxas de codificação* r_m *são iguais a* r*, e que todas as variáveis* U^m*,* $m \in \{1, 2, .., \mathrm{M}\}$ *estão igualmente distribuídas de modo que a entropia conjunta depende só do número das fontes envolvidas e da correlação entre qualquer par de fontes. Neste caso se deduz pela simetria do exemplo que todos os valores* R_m *são iguais a um valor comum* R*. Assim, da equação (2.9) se obtém*

$$\frac{\mathrm{C}}{r} \geq \mathrm{R}. \tag{2.39}$$

Conhecendo do teorema de $Slepian - Wolf$ *podemos escrever que*

$$\begin{array}{rcl} \mathrm{R} & \geq & \frac{H(U^1 U^2 ... U^{\mathrm{M}})}{\mathrm{M}} \\ & \geq & \frac{H(\mathrm{M},\rho)}{\mathrm{M}}, \end{array} \tag{2.40}$$

onde $H(\mathrm{M}, \rho)$ *indica a entropia de* M *fontes igualmente distribuídas com uma probabilidade de erro* ρ *com uma fonte comum de símbolos equiprováveis. Das equações anteriores então se deduz a seguinte equação de restrição para os possíveis valores de* R

$$\frac{\mathrm{C}}{r} \geq \mathrm{R} \geq \frac{H(\mathrm{M}, \rho)}{\mathrm{M}}. \tag{2.41}$$

Se não é verdadeiro que $\frac{\mathrm{C}}{r} \geq \frac{H(\mathrm{M},\rho)}{\mathrm{M}}$*, então não é possível uma comunicação sem ruído para um valor de taxa de codificação* r*. No caso de cumprir-se a desigualdade se assume que o valor de* R *é igual a*

$$\mathrm{R} = \frac{H(\mathrm{M}, \rho)}{\mathrm{M}}. \tag{2.42}$$

Assim, o limite virtual de Shannon para um canal BI-AWGN é

$$(\hat{E}_b/N_0)_{BI-AWGN} = \frac{1}{2r\mathrm{R}\ \phi^{-1}(r\mathrm{R})} \tag{2.43}$$

e o limite virtual de Shannon para um canal hard-decision BI-AWGN é

$$(\hat{E}_b/N_0)_{HD-BI-AWGN} = \frac{\{Q^{-1}(h^{-1}(1-r\mathrm{R})\}^2}{2r\mathrm{R}}. \tag{2.44}$$

É fácil perceber que se ρ tende a zero então $\mathrm{R} = 1/\mathrm{M}$. *O valor de ρ próximo de zero indica que na verdade o que se está fazendo é enviar* M *vezes a mesma informação. Isto pode ser interpretado como que se houvesse um só canal de comunicação com uma taxa de codificação de canal igual a r/M.*

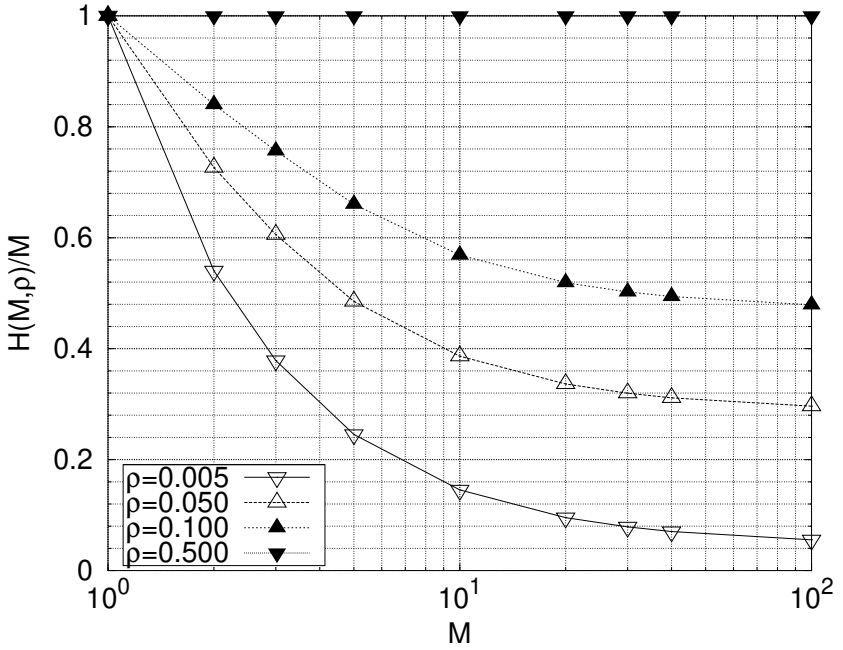

Figura 2.12: $H(\mathrm{M},\rho)/\mathrm{M}$ para distintos valores de ρ e M

A Figura 2.12 mostra o valor $\mathrm{R} = H(\mathrm{M},\rho)/\mathrm{M}$ *para distintos valores de ρ e* M. *Pode-se ver que para um ρ fixo e* M *tendendo a infinito as curvas tem um valor assintótico.*

Exemplo 2.4.2 (Limite de um canal *HD BI-AWGN* com $r = 1/2$ e $r = 2/3$) *Neste caso as M fontes tem o mesmo modelo de correlação que no Exemplo 2.3.2, onde a probabilidade de erro de bit dos canais BSC do modelo são iguais a ρ e a taxa de codificação fonte-canal tem um valor de r.*

A Figura 2.13 mostra o comportamento do limite virtual de Shannon para um sistema com canais do tipo hard-decision BI-AWGN, o limite e desenhado para uma taxa de codificação $r = 1/2$, um $\rho = \{0.005, 0.05, 0.1, 0.5\}$ e $M = \{1, 2, 3, 5, 10, 20, 30, 40, 100\}$. A Figura 2.14 mostra o comportamento do limite virtual de Shannon para um sistema com canais do tipo hard-decision BI-AWGN, com os mesmos parâmetros que a figura anterior com a diferença que a taxa de codificação fonte-canal $r = 2/3$.

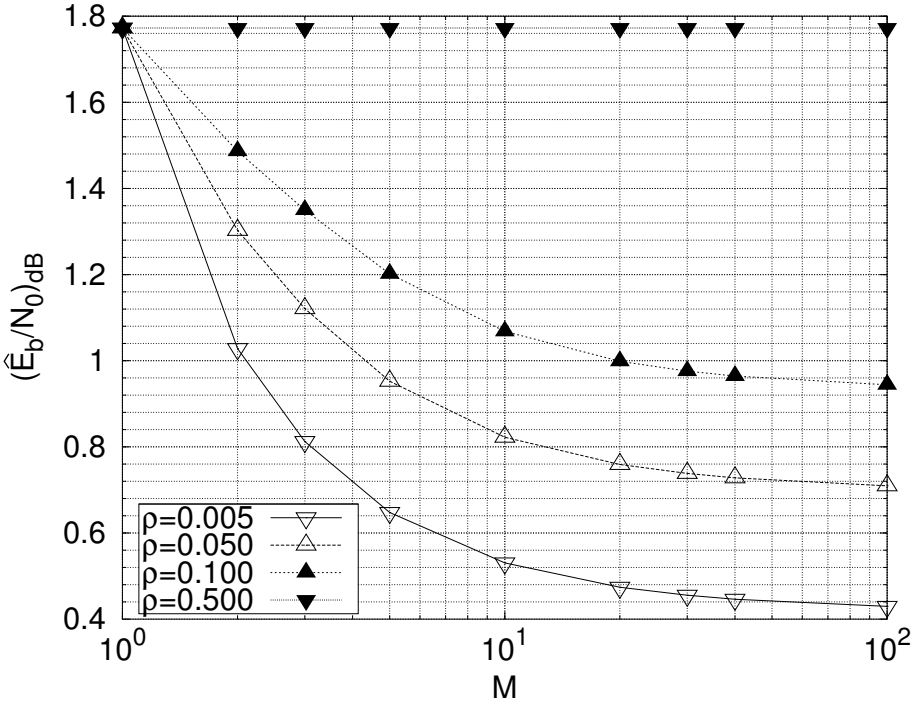

Figura 2.13: Limite virtual de Shannon para uma taxa $r = 1/2$.

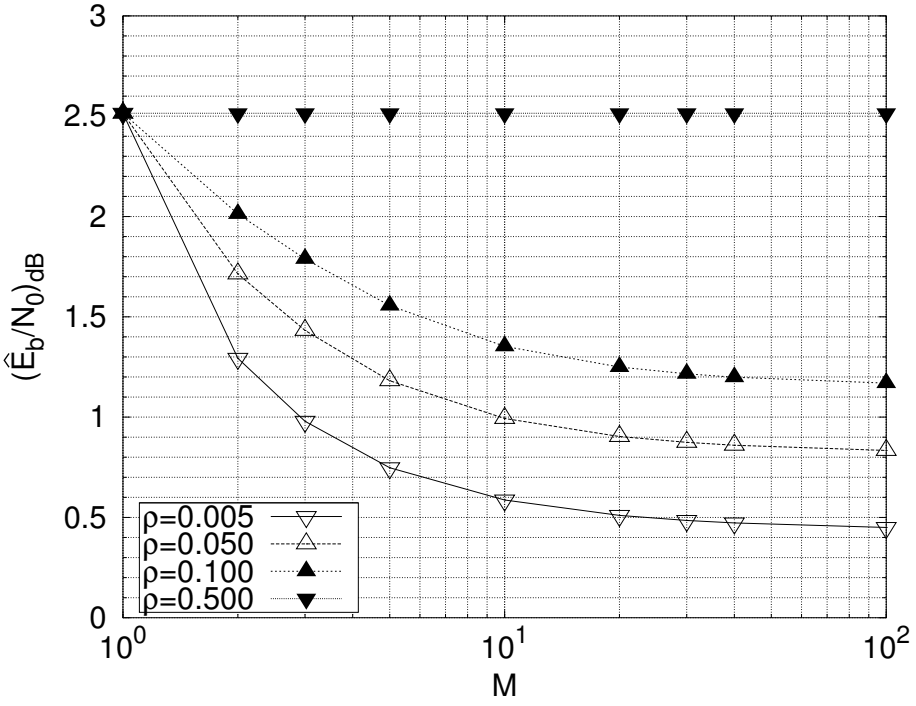

Figura 2.14: Limite virtual de Shannon para uma taxa $r = 2/3$.

Capítulo 3

Decodificação Conjunta

Neste capítulo duas contribuições serão apresentadas. A primeira é a definição de uma variante simplificada do algoritmo *Parallel Weighted Bit − Flipping* (Wu, Zhao & You 2007). Isto pode ser visto na Seção 3.2.5. A segunda contribuição é a proposição de um algoritmo para a decodificação conjunta em sistemas de transmissão com múltiplas fontes quando estas possuem correlação. Este algoritmo pode ser visto na Seção 3.3.

3.1 Introdução

Existem muitos algoritmos simples (baixa complexidade computacional) para realizar uma decodificação da informação que percorre um canal com ruído. Estes algoritmos podem usar uma decisão suave (do inglês, soft decision), como os algoritmos *Weighted Bit − Flipping* (Kou, Lin & Fossorier 2001) e *Parallel Weighted Bit − Flipping* (Wu et al. 2007), ou abrupta (do inglês, hard decision) como os algoritmos *Bit − Flipping* (Gallager 1962) e *Parallel Hard Bit − Flipping* (Pujaico Rivera 2011). Se diz que um algoritmo de decodificação realiza uma decisão suave quando se utiliza na sua predição números reais como dados de entrada, pelo contrário se diz que o algoritmo de decodificação realiza uma decisão abrupta quando usa dados inteiros.

Os algoritmos de decodificação estudados neste capítulo usaram códigos *LDPC*. Todas as fontes usaram a mesma taxa de codificação fonte-canal r = K/N.

Como é ilustrado na Figura 2.1, se tem M fontes que passam por M canais com ruído. Na saída dos canais ruidosos tem-se M vetores linha $\mathbf{Y}^m$, $\forall m \in \{1, 2, ..., \mathrm{M}\}$, e sua contraparte abrupta $\mathbf{Z}^m$. Pode-se realizar com estes vetores uma decodificação para obter uma estimação dos vetores linha $\hat{\mathbf{U}}^m$ de $\mathbf{U}^m$, baseando-se só na redundância acrescentada por cada codificador fonte-canal de taxa r: a isto chama-se decodificação independente. Ou pode-se usar a informação redundante acrescentada pelo conhecimento da informação ruidosa recebida das outras M − 1 fontes $\mathbf{U}^m$: a isto chama-se decodificação conjunta.

3.2 Algoritmos de Decodificação Independente

Nesta seção se descrevem os algoritmos sem o uso do índice m dado que não existe aproveitamento da redundância da informação para corrigir erros. Assim os vetores $\mathbf{U}^m$, $\mathbf{Y}^m$ e $\mathbf{Z}^m$ passam a ser $\mathbf{U}$, $\mathbf{Y}$ e $\mathbf{Z}$, respectivamente.

3.2.1 Algoritmo Bit-Flipping

O algoritmo de decodificação $Bit-Flipping$ (BF) foi idealizado por Gallager em (Gallager 1962) como uma proposta para identificar e corrigir os erros na transmissão de informação quando um vetor $\mathbf{U}$ codificado em $\mathbf{V}$ transita através de um canal com ruído. Na saída do canal se tem um novo vetor binário chamado $\mathbf{Z}$. Para corrigir possíveis erros em $\mathbf{Z}$, o algoritmo atribui a cada bit z_n, $\forall n \in \{0, 1, .., \mathrm{N}-1\}$ de $\mathbf{Z}$ uma confiabilidade i_n pertencente ao vetor $\mathbf{I}$. Esta confiabilidade indica com seus valores negativos ou positivos a menor ou maior probabilidade de que esse bit esteja errado.

Para a decodificação é necessário calcular o vetor de síndrome $\mathbf{S}$ de L bits s_l, $\forall l \in \{0, 1, .., \mathrm{L}-1\}$ e conhecer a matriz de verificação de paridade $\mathbf{H}$. A matriz $\mathbf{H}$ é composta por L vetores binários h_l de comprimento N. $\mathbf{S}$ pode ser calculado com

$$\mathbf{S} = \mathbf{Z}\ \mathbf{H}^t, \tag{3.1}$$

onde $\mathbf{H}^t$ é a transposta de $\mathbf{H}$.

Definimos $\mathcal{M}(n)$ como o conjunto de todos os valores que l pode assumir com $h_{l,n} = 1$, ou seja,

$$\mathcal{M}(n) = \{l | h_{l,n} = 1\}. \tag{3.2}$$

O critério para atribuir as confiabilidades i_n do vetor $\mathbf{I}$ segue a seguinte equação

$$i_n = \sum_{l \in \mathcal{M}(n)} s_l. \tag{3.3}$$

Pode-se entender melhor $\mathcal{M}(n)$ mediante a Figura 3.1 onde:

- $\mathcal{M}(1) = \{1, 2\}$, representa o conjunto dos índices dos bits s_l que estão ligados ao bit z_1.
- $\mathcal{M}(2) = \{0, 1\}$, representa o conjunto dos índices dos bits s_l que estão ligados ao bit z_2.

Após estabelecer o critério de geração de confiabilidades i_n para cada bit z_n, é necessário gerar também um critério para trocar os bits. O algoritmo BF troca todos os bits com $i_n \geq \delta$. Usa-se como limiar δ o valor máximo de i_n no vetor $\mathbf{I}$ gerado seguindo a equação (3.3). O Algoritmo 1 descreve a implementação do algoritmo BF.

A decodificação BF pode-se entender como um algoritmo baseado em confiabilidades, onde o conjunto de bits menos confiáveis são considerados errados e seu valor é trocado. A confiabilidade de cada bit z_n é calculada somando a quantidade de bits de verificação de paridade s_l que ligados a z_n indiquem a existência de erro. *Muito parecido a uma votação, onde só os votos em contra são contabilizados para o bit em estudo. Ao final, o conjunto de bits que tenham mais votos em contra serão trocados.*

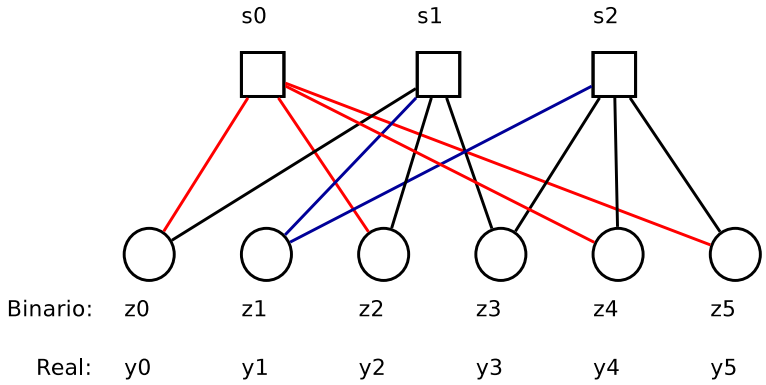

Figura 3.1: Gráfico de Tanner de um código binário (N,K)=(6,4).

Algoritmo 1 Decodificação dos algoritmos da família $Bit - Flipping$.

1. Inicia-se com $j = 0$ e o vetor de decisão abrupta $\mathbf{Z} = hard(\mathbf{Y})$, $\mathbf{Z}^{(j)} = (z_0^{(j)}, \cdots, z_n^{(j)}, \cdots, z_{\mathrm{N}-1}^{(j)}) = \mathbf{Z}$.
2. Calcule a síndrome $\mathbf{S} = \mathbf{Z}^{(j)}\mathbf{H}^t \pmod 2$. Se todos os valores s_l com $l = \{0, \cdots, \mathrm{L}-1\}$ são zero, então para-se a decodificação e se retorna o vetor $\mathbf{Z}^{(j)}$.
3. Para $n = \{0, \cdots, \mathrm{N}-1\}$, se calcula a função de "flipping" i_n.
4. Identifica-se o conjunto $\{n^*\}$, onde $n^* = \arg_n \max i_n$.
5. Troca-se de valor todos os bits $z_n^{(j)} \in \mathbf{Z}^{(j)}$ onde $n \in \{n^*\}$, e faz-se $\mathbf{Z}^{(j+1)} = \mathbf{Z}^{(j)}$.
6. Se o número máximo de iterações não é atingido, faz-se $j = j + 1$ e continua-se no passo 2. Caso contrário se detém a decodificação e se retorna $\mathbf{Z}$.

É importante ressaltar que o Algoritmo 1 é a forma geral que tomará todos os algoritmos da família $Bit - Flipping$. As mudanças nestes algoritmos acontecerão na implementação da função de "flipping" i_n do passo 3.

3.2.2 Algoritmo Parallel Hard Bit-Flipping

O algoritmo de decodificação $Parallel\ Hard\ Bit - Flipping$ $(PHBF)$, proposto em (Pujaico Rivera 2011), trabalha sobre um vetor de entrada binário $\mathbf{Z}$ e o vetor de síndrome $\mathbf{S}$ calculado com a equação (3.1). O algoritmo $PHBF$ tem como regra de geração do vetor de confiabilidades $\mathbf{I}$,

$$i_n = \sum_{l \in \mathcal{M}(n)} (2s_l - 1), \tag{3.4}$$

onde i_n, $\forall n \in \{0, 1, ..., \mathrm{N}-1\}$ é um elemento do vetor $\mathbf{I}$. O critério de troca dos bit errados é semelhante ao algoritmo BF. Troca-se todos os bits com um $i_n \geq \delta$. Para a implementação atual usou-se como δ o maior valor de i_n. O Algoritmo 1 detalha todo o procedimento para a decodificação $PHBF$, usando para o passo 3 a equação (3.4).

Existe uma relação entre os algoritmos BF e $PHBF$ pois ambos utilizam um regra de confiabilidade parecida. A primeira vista poderia parecer que a confiabilidade de $PHBF$ (veja equação 3.4) é simplesmente uma combinação linear da confiabilidade de BF (veja equação 3.3). Pelo qual não deveria existir nenhuma diferença quando se acha a confiabilidade do valor máximo, e portanto também a decodificação de $\mathbf{Z}$. Isto é verdade só no caso em que o número de elementos de $\mathcal{M}(n)$ é constante para todo $n \in \{0, 1, \ldots, \mathrm{N}-1\}$. A verificação disto é fácil de ver se consideramos a quantidade de bits por coluna de $\mathbf{H}$ como $X(n)$, onde

$$X(n) = \sum_{l=0}^{\mathrm{L}-1} h_{l,n}. \tag{3.5}$$

$X(n)$ também pode ser interpretado como a quantidade de bits de verificação de paridade de $\mathbf{S}$ que estão conectados à z_n.

Se considerarmos que se tem c bits de verificação de paridade errados para o bit z_n então o valor da confiabilidade segundo o algoritmo BF seria $\{i_n\}_{BF} = c$. Por outro lado, a confiabilidade para o algoritmo $PHBF$ seria $\{i_n\}_{PHBF} = (+1)c + (-1)(X(n) - c) = 2c - X(n)$. Unindo ambas ideias, obteríamos a equação

$$\{i_n\}_{PHBF} = 2\{i_n\}_{BF} - X(n). \tag{3.6}$$

A equação (3.6) mostra claramente que se todos os bits z_n possuem a mesma quantidade de bits de verificação de paridade, isto é se a matriz $\mathbf{H}$ possui um grau de coluna constante, os algoritmos BF e $PHBF$ levam ao mesmo resultado na decodificação. No caso em que o grau não é constante os algoritmos tem desempenhos distintos.

A confiabilidade de cada bit z_n é calculada somando a quantidade de bits de verificação de paridade em $\mathbf{S}$ que ligados a z_n indiquem a existência de um erro, e diminuindo a quantidade de bits de verificação de paridade em $\mathbf{S}$ que ligados ao bit z_n não indiquem a existência de erro. *Muito parecido a uma votação, onde são contabilizados os votos em contra e os votos a favor que anulam um voto em contra, ao final o conjunto de bits que tenham mais votos em contra serão trocados.*

O desempenho do algoritmo $PHBF$ para o caso de matrizes de verificação de paridade H de tipo $LDGM$ pode ser visto na Seção 3.4 nas Figuras 3.4, 3.5, 3.7, 3.9, 3.11, 3.12, 3.13. E no caso de matrizes de tipo $LDPC$ na Figura 3.14.

A Figura 3.2 mostra um diagrama de fluxo do algoritmo $PHBF$. Foi incluído nesta figura o índice "m" que indica em que canal se realiza a decodificação. Este diagrama de fluxo pode ser usado para todos os algoritmos da família $Bit-Flipping$, com a diferença de que cada variante $Bit-Flipping$ usa um diferente procedimento para obter o vetor de confiabilidades.

3.2.3 Algoritmo Weighted Bit-Flipping

O algoritmo $Weighted\ Bit{-}Flipping$ (WBF) foi proposto em (Kou et al. 2001). O algoritmo trabalha sobre um vetor de entrada real $\mathbf{Y}$ e o vetor de síndromes $\mathbf{S}$. Para obter o vetor de confiabilidades $\mathbf{I}$ em ponto flutuante, com elementos i_n, $\forall n \in \{0, 1, ..., \mathrm{N}-1\}$ calcula-se

$$|w_l| = \min_{j \in \mathcal{N}(l)} |y_j|, \tag{3.7}$$

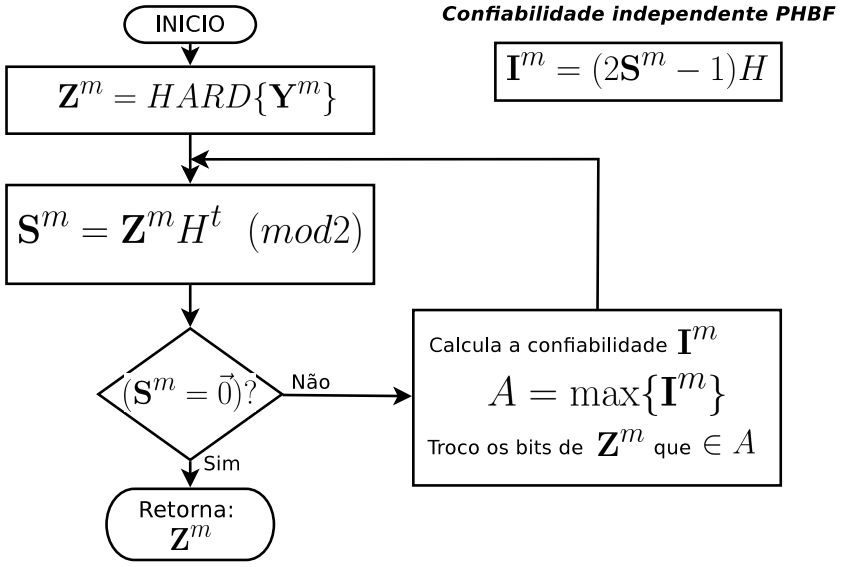

Figura 3.2: Algoritmo Parallel Hard Bit-Flipping.

$$i_n = \sum_{l \in \mathcal{M}(n)} (2s_l - 1)|w_l|. \tag{3.8}$$

Define-se $\mathcal{N}(l)$ como o conjunto de todos os valores que pode tomar n tal que $h_{l,n} = 1$, ou seja,

$$\mathcal{N}(l) = \{n | h_{l,n} = 1\}. \tag{3.9}$$

Pode-se entender melhor $\mathcal{N}(l)$ mediante a Figura 3.1 onde:

- $\mathcal{N}(0) = \{0, 2, 4, 5\}$, representa o conjunto dos índices dos bits z_n que estão ligados ao bit s_0.
- $\mathcal{N}(1) = \{0, 1, 2, 3\}$, representa o conjunto dos índices dos bits z_n que estão ligados ao bit s_1.

O critério de troca dos bit errados é semelhante ao algoritmo BF. Troca-se todos os bits com um $i_n \geq \delta$. Para a implementação atual usou-se como δ o maior valor de i_n. O Algoritmo 1 detalha todo o procedimento para a decodificação WBF, usando para o passo 3 a equação (3.8).

O desempenho do algoritmo WBF para o caso de matrizes de verificação de paridade de tipo $LDGM$ pode ser visto na Seção 3.5 na Figura 3.16.

3.2.4 Algoritmo Parallel Weighted Bit-Flipping

O algoritmo $Parallel\ Weighted\ Bit-Flipping$ ($PWBF$) foi proposto em (Wu et al. 2007) como uma modificação do algoritmo $Improved\ Modified\ Weighted\ Bit-Flipping$ ($IMWBF$) proposto em (Jiang, Zhao, Shi & Chen 2005) que a sua vez foi uma melhora do algoritmo $Weighted\ Bit-Flipping$ (WBF). O algoritmo trabalha sobre um vetor de entrada real $\mathbf{Y}$ e o vetor de síndromes $\mathbf{S}$. Para obter o vetor de confiabilidades $\mathbf{I}$ em ponto fixo, com elementos

i_n, $\forall n \in \{0, 1, ..., \mathrm{N}-1\}$. Primeiro obtém-se o vetor de confiabilidades $\mathbf{F}$ com elementos f_n em ponto flutuante seguindo o critério do algoritmo $IMWBF$.

$$|w_{n,l}| = \min_{j \in \mathcal{N}(l) \backslash n} |y_j|, \tag{3.10}$$

$$f_n = \sum_{l \in \mathcal{M}(n)} (2s_l - 1)|w_{n,l}| - \alpha |y_n|, \tag{3.11}$$

agora com f_n, $n \in \{0, 1, \ldots, \mathrm{N}-1\}$ calcula-se uma confiabilidade de erro de bit em ponto fixo.

Iniciando $\mathbf{I} = 0$, para todo $s_l = 1$ calcula-se:

$$i_{\theta_l} \leftarrow i_{\theta_l} + 1, \tag{3.12}$$

$$\theta_l = \arg\{\max_{j \in \mathcal{N}(l)} f_j\}, \tag{3.13}$$

onde $l \in \{0, 1, \ldots, \mathrm{L}-1\}$ e $\theta_l \subset \{0, 1, \ldots, \mathrm{N}-1\}$, O critério de troca dos bit errados é semelhante ao algoritmo BF. Troca-se todos os bits com um $i_n \geq \delta$. Para a implementação atual usou-se como δ o maior valor de i_n. O Algoritmo 1 detalha todo o procedimento para a decodificação $PWBF$, usando para o passo 3 as equações (3.12) e (3.13).

3.2.5 Algoritmo Parallel Weighted Bit-Flipping Simplificado

Este algoritmo $SPWBF$ é proposto nesta tese como uma variante do algoritmo WBF e $PWBF$. Este foi modificado com a finalidade de simplificar e melhorar o desempenho do algorítimo na decodificação conjunta, também foi retirado o parâmetro α do algoritmo $PWBF$ de modo que o critério de geração de confiabilidades é semelhante ao algoritmo WBF. O algoritmo trabalha sobre um vetor de entrada real $\mathbf{Y}$. Para obter o vetor de confiabilidades $\mathbf{I}$ com elementos i_n em ponto fixo. Primeiro obtém-se o vetor de confiabilidades $\mathbf{F}$ com elementos f_n em ponto flutuante

$$|w_l| = \min_{j \in \mathcal{N}(l)} |y_j|, \tag{3.14}$$

$$f_n = \sum_{l \in \mathcal{M}(n)} (2s_l - 1)|w_l|, \tag{3.15}$$

agora com f_n, $n \in \{0, 1, \ldots, \mathrm{N}-1\}$, calcula-se uma confiabilidade de erro de bit em ponto fixo. Inicia-se o vetor de confiabilidades $\mathbf{I} = 0$ e:

Para todo $s_l = 1$ calcula-se:

$$i_j \leftarrow i_j + 1, \quad \forall\ j \in \{\mathcal{N}(l)/s_l = 1\} \tag{3.16}$$

$$i_{\theta_l} \leftarrow i_{\theta_l} + 1, \tag{3.17}$$

$$\theta_l = \arg\{\max_{j \in \{\mathcal{N}(l)/s_l = 1\}} f_j\}. \tag{3.18}$$

Para todo $s_l = 0$ calcula-se:

$$i_j \leftarrow i_j - 1, \quad \forall\ j \in \{\mathcal{N}(l)/s_l = 0\} \tag{3.19}$$

$$i_{\theta_l} \leftarrow i_{\theta_l} - 1, \tag{3.20}$$

$$\theta_l = \arg\{\min_{j \in \{\mathcal{N}(l)/s_l=0\}} f_j\}, \tag{3.21}$$

onde $l \in \{0, 1, \ldots, \mathrm{L}-1\}$ e $\theta_l \subset \{0, 1, \ldots, \mathrm{N}-1\}$. O critério de troca dos bit errados é semelhante ao algoritmo BF. Troca-se todos os bits com um $i_n \geq \delta$. Para a implementação atual usou-se como δ o maior valor de i_n. O Algoritmo 1 detalha todo o procedimento para a decodificação $PWBF$ simplificado, usando para o passo 3 as equações (3.16) e (3.21).

O desempenho do algoritmo $SPWBF$ para o caso de matrizes de verificação de paridade de tipo $LDGM$ pode ser visto na Seção 3.5 na Figura 3.16.

3.3 Algoritmos de Decodificação Conjunta

Os algoritmos de decodificação conjunta aqui apresentados são modificações dos algoritmos de decodificação independente $Parallel\ Hard\ Bit - Flipping$ e $Parallel\ Weighted\ Bit - Flipping$ apresentados nas seções anteriores. Em geral a modificação nos algoritmos independentes será trocar o uso da confiabilidade independente i_n^m do n-ésimo bit z_n^m da m-ésima fonte por uma confiabilidade total ${e_T}_n^m \in \mathbf{E}_T$, $\forall n \in \{0, 1, .., \mathrm{N}-1\}$ e $\forall m \in \{1, 2, .., \mathrm{M}\}$, onde

$$\mathbf{E}_T{}^m = \mathbf{I}^m + \beta \mathbf{E}^m, \tag{3.22}$$

$${e_T}_n^m = i_n^m + \beta e_n^m. \tag{3.23}$$

β é um fator de ponderação e $\mathbf{E}^m$ é a confiabilidade conjunta para cada bit z_n^m do vetor recebido $\mathbf{Z}^m$. Esta confiabilidade conjunta será calculada usando as confiabilidades independentes dos bits recebidos dos outros $\mathrm{M}-1$ canais e ponderando-os em função da correlação entre estes. Pode-se ver facilmente que o caso de decodificação independente é um caso especifico da decodificação conjunta quanto $\beta = 0$.

3.3.1 Algoritmo em Ponto Fixo para Fontes Distribuídas

O algoritmo aqui apresentado é uma modificação para o caso de fontes distribuídas dos algoritmos com o vetor de confiabilidades $\mathbf{I}$ em ponto fixo. Neste algoritmo o vetor $\mathbf{I}$ é trocado pelo vetor em ponto fixo $\mathbf{E}_T{}^m$.

No caso de usar matrizes $\mathbf{G}$ e $\mathbf{H}$ de tipo $LDGM$, $\mathbf{E}_T{}^m$ é descrito como

$$\mathbf{E}_T{}^m = \mathbf{I}^m + \lfloor \beta\ \mathbf{E}^m \rfloor, \tag{3.24}$$

onde $\lfloor b \rfloor$ é o maior inteiro menor ou igual a b, e β é um fator de ponderação entre a confiabilidade independente e a confiabilidade conjunta, que atua sobre todas as confiabilidades dos bits da palavra codificada recebida $\mathbf{Z}$.

No caso de usar-se uma matriz $\mathbf{H}$ de tipo $LDPC$ e uma matriz $\mathbf{G}$ de tipo $LDGM$, β só atua sobre as confiabilidades dos bits de informação da palavra codificada $\mathbf{Z}$, para todos os outros bits β assume o valor nulo. Para entender isto tem que se ressaltar que todos os bits de paridade são resultados de passar o vetor de informação por uma matriz $LDPC$. Como é visto

no Apêndice A.5, a correlação vai perdendo-se com o aumento do número de uns por coluna no cálculo da paridade na matriz geradora G.

Para calcular $\mathbf{E}^m$ utiliza-se

$$\lambda_{m,a,n} = \begin{cases} -1 & , se \quad z_n^m \neq z_n^a \\ 0 & , se \quad m = a \\ +1 & , se \quad z_n^m = z_n^a \end{cases}, \tag{3.25}$$

$$e_n^m = \frac{1}{\mathrm{M}-1}\left(\sum_{a=1}^{\mathrm{M}} \lambda_{m,a,n}\ i_n^a\ corr(m,a)\right), \tag{3.26}$$

onde $corr(m,a)$ é a correlação entre a a-ésima fonte e a m-ésima fonte. Neste capítulo, as correlações serão assumidas como conhecidas. É importante ressaltar que se analisamos os algoritmos independentes em ponto fixo como $PHBF$ e $PWBF$, se verá que não é calculado o vetor $\mathbf{I}^m$ se o vetor de síndrome $\mathbf{S}^m$ é nulo. Para implementar corretamente o algoritmo para fontes distribuídas é necessário modificar os algoritmos independentes para que mesmo quando a síndrome seja nula se calcule o vetor de confiabilidades $\mathbf{I}^m$. Assim os algoritmos independentes são modificados obtendo-se o Algoritmo 2 para fontes distribuídas.

Algoritmo 2 Algoritmo em ponto fixo para fontes distribuídas

1. Inicia-se com $j = 0$ e o vetor de decisão abrupta $\mathbf{Z}^m = hard(\mathbf{Y}^m)$, $\mathbf{Z}^{m(j)} = (z_0^{m(j)}, \cdots, z_n^{m(j)}, \cdots, z_{\mathrm{N}-1}^{m}{}^{(j)}) = \mathbf{Z}^m$.
2. Calcula-se a síndrome $\mathbf{S}^m = \mathbf{Z}^{m(j)}\mathbf{H}^t \pmod 2$ e as confiabilidades i_n^m. Se o vetor $\mathbf{S}^m$ é nulo, então para a decodificação e se retorna os vetores $\mathbf{Z}^{m(j)}$ e $\mathbf{I}^m$.
3. Para $n = \{0, \cdots, \mathrm{N}-1\}$, avalia-se a função de flipping $e_{T}{}_n^m$ para obter o vetor $\mathbf{E}_T{}^m$.
4. Identifica-se o conjunto $\{n^*\}$, onde $n^* = \arg_n \max i_n^m$.
5. Troca-se de valor todos os bits $z_n^{m(j)}$ onde $n \in \{n^*\}$, e faz-se $\mathbf{Z}^{m(j+1)} = \mathbf{Z}^{m(j)}$.
6. Se o máximo número de iterações não é atingido, faz-se $j = j+1$ e vai-se ao passo 2. Caso contrário, se detém a decodificação e se retorna $\mathbf{Z}^m$.

Usando o Algoritmo 2 com o critério de geração de confiabilidades do algoritmo $PHBF$ se obtém um algoritmo que passará a ser chamado de $PHBF$ distribuído ($DPHBF$). O desempenho deste algoritmo pode ser visto na Seção 3.4. Se o Algoritmo 2 usa o critério de geração de confiabilidades do algoritmo $SPWBF$ se obtém um algoritmo que passará a ser chamado de $SPWBF$ distribuído ($DSPWBF$). O desempenho deste algoritmo pode ser visto na Seção 3.5.

A Figura 3.3 mostra um diagrama de fluxo do algoritmo $DPHBF$. Este diagrama de fluxo pode ser usado para representar também ao algoritmo $DSPWBF$, com a diferença de que deveria ser mudado o procedimento para obter o vetor de confiabilidades $\mathbf{I}^m$.

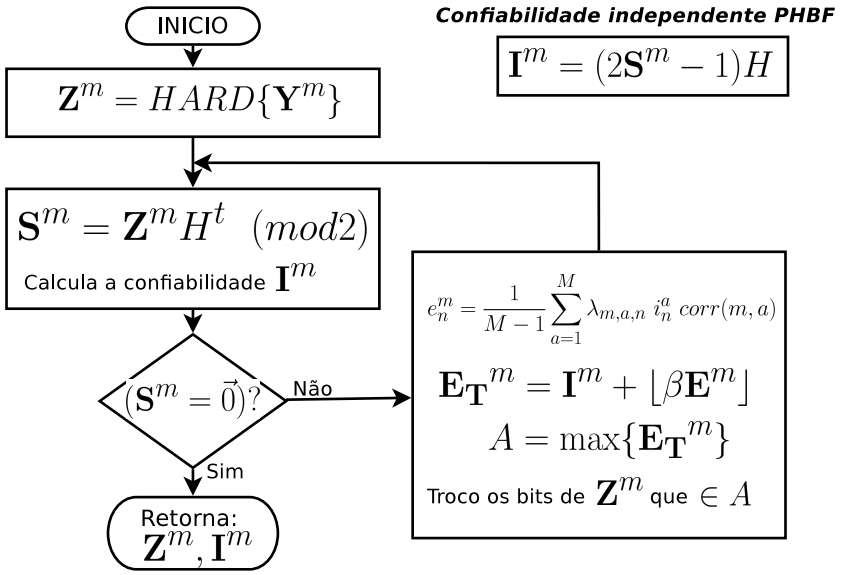

Figura 3.3: Algoritmo Parallel Hard Bit-Flipping Distribuído.

3.4 Desempenho dos Algoritmos $PHBF$ e $DPHBF$

Todas as simulações apresentadas nesta tese usam o modelo de geração de fontes correlacionadas como na Figura 1.10, com $P(u^0 = 1) = 0.5$.

3.4.1 Algoritmo $DPHBF$ com Matrizes $LDGM$

Estudo do Desempenho do Parâmetro β

As Figuras 3.4, 3.5, 3.6, 3.7 e 3.8 mostram o desempenho dos algoritmos $PHBF$ e $DPHBF$ para matrizes $\mathbf{G}$ e $\mathbf{H}$ de tipo $LDGM$, a quantidade de uns por coluna de $\mathbf{H}$ ligados aos bits de informação $\mathcal{X}$ varia entre 8 e 9. O código tem uma taxa r = 2/3, com N = 3000 e K = 2000. A quantidade de iterações máximas dos algoritmos é igual a $IT = 60$.

A Figura 3.4 mostra o desempenho do algoritmo $PHBF$, desenhado com "$*$", e o algoritmo $DPHBF$ para distintos valores de fator de ponderação β da confiabilidade conjunta, sendo os valores que toma $\beta = \{0.3, 0.5, 0.7\}$. Todas as fontes U^m, $\forall m \in \{1, 2, 3\}$ desta simulação tem uma probabilidade de erro $\rho = 0.1$ com a fonte escondida U^0. Isto significa que duas fontes U^m quaisquer tem uma correlação de $corr(U^i, U^j) = 0.64$, $\forall i \neq j$, ver equação (1.65). Também esta figura mostra o desempenho de um canal não codificado legendado como "Uncoded" e o limite inferior do desempenho do BER para uma matriz $\mathbf{H}$ de tipo $LDGM$, desenhado com "$\circ$". Pode-se ver que quando o fator β cresce o desempenho melhora em valores baixos de E_b/N_0 e que quando se diminui o valor de β o desempenho para valores altos de E_b/N_0 melhora. Em qualquer caso deve-se obter experimentalmente qual será o melhor valor de β para cada valor de E_b/N_0. Tendo em conta isto gera-se a Figura 3.5, que mostra o desempenho do algoritmo $DPHBF$ usando distintos valores de β para cada E_b/N_0. A Figura 3.5 mostra uma melhora no desempenho do algoritmo $DPHBF$ em relação ao algoritmo $PHBF$ de aproximadamente $0.4dB$.

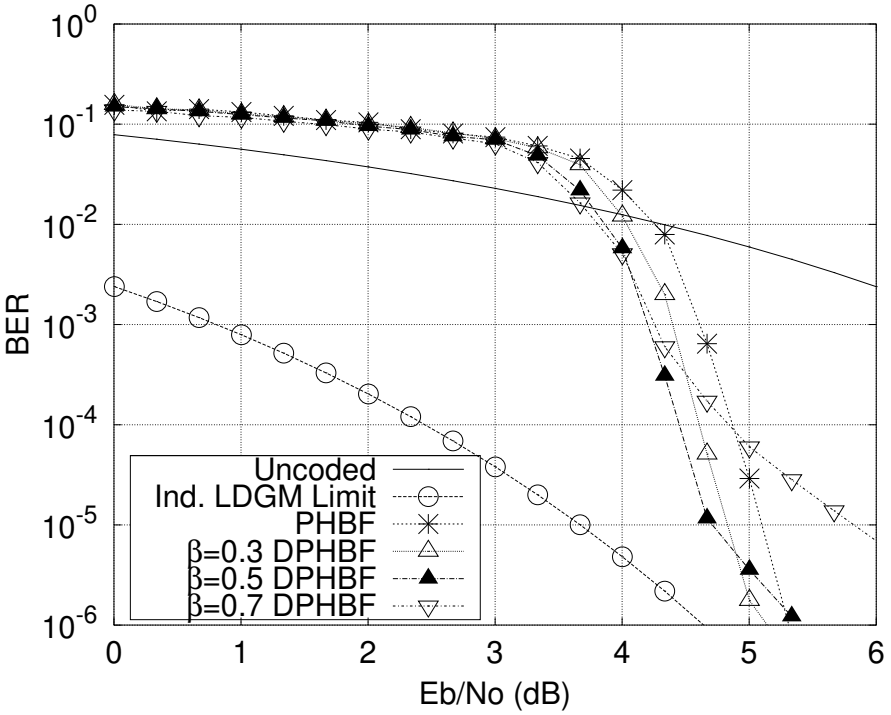

Figura 3.4: BER usando os algoritmos $PHBF$ e $DPHBF$ com uma matriz de tipo $LDGM$, $\mathcal{X} = \{8, 9\}$, K = 2000, N = 3000, $\rho = 0.1$ e M = 3 para distintos valores de beta.

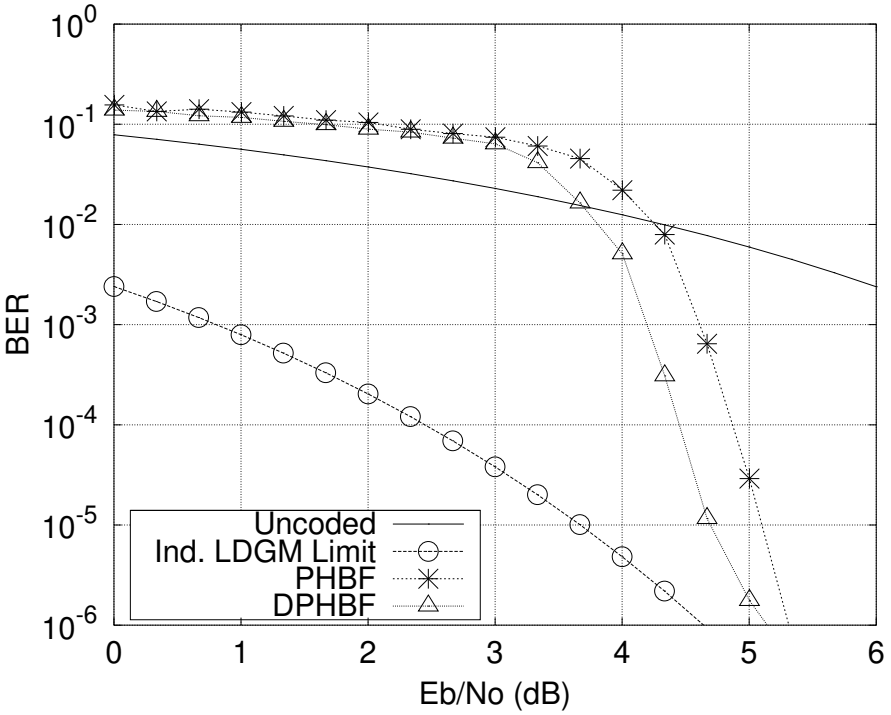

Figura 3.5: BER usando os algoritmos $PHBF$ e $DPHBF$ com uma matriz de tipo $LDGM$, $\mathcal{X} = \{8, 9\}$, K = 2000, N = 3000, $\rho = 0.1$ e M = 3 quando varia β.

O gráfico da Figura 3.6 mostra que valor de β foi usado para cada valor de E_b/N_0 na Figura 3.5, o valor de β tem um valor máximo de $\beta = 0.7$ após o qual não pode-se obter melhora nenhuma do algoritmo $DPHBF$.

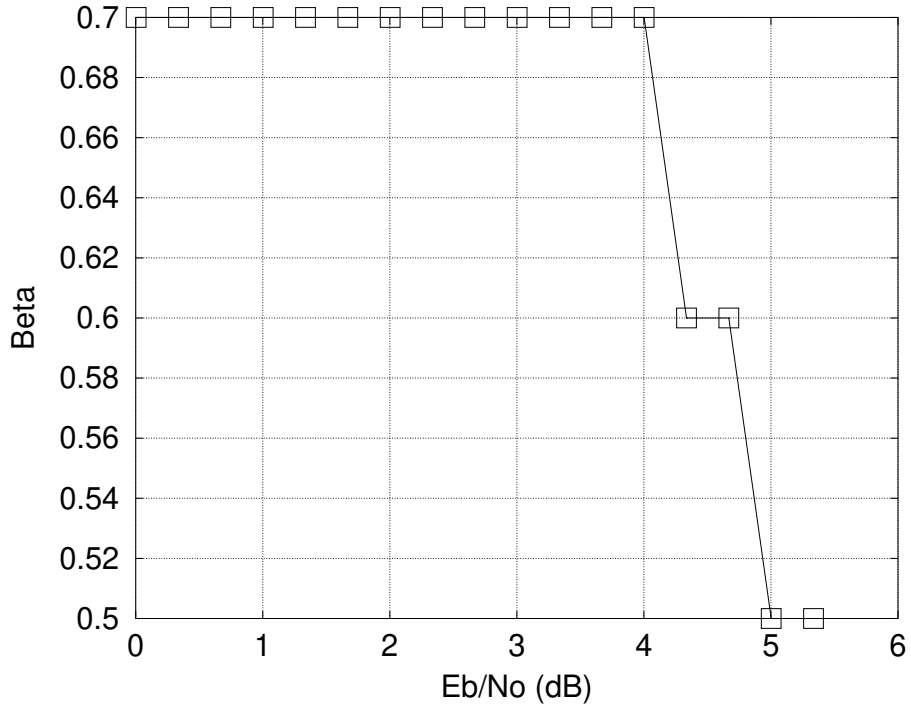

Figura 3.6: Valor de β no desempenho dos algoritmos $PHBF$ e $DPHBF$ com uma matriz de tipo $LDGM$, $\mathcal{X} = \{8, 9\}$, K = 2000, N = 3000, $\rho = 0.1$ e M = 3.

As Figuras 3.7 e 3.8 mostram o desempenho para um modelo de sistema semelhante ao usado na Figura 3.5, com a diferença que se tem uma probabilidade de erro ρ com a fonte escondida U^0 igual a $\rho = 0.005$, isto é, duas fontes quaisquer tem uma correlação $corr(U^i, U^j) = 0.9801$, $\forall i \neq j$. Para um valor $\beta > 0.7$ não se observam melhoras significativas no algoritmo $DPHBF$ em relação ao algoritmo $PHBF$. O algoritmo $DPHBF$ tem um ganho de aproximadamente de $1.0dB$ em relação ao algoritmo $PHBF$. Pode-se ver como a decodificação conjunta sobrepassa o limite inferior do desempenho independente para matrizes $LDGM$.

As Figuras 3.9 e 3.10 mostram o desempenho dos algoritmos $PHBF$ e $DPHBF$ para matrizes **G** e **H** de tipo $LDGM$, a quantidade de uns por coluna de **H** ligados aos bits de informação é de $\mathcal{X} = 9$. O código tem uma taxa r = 1/2, com N = 30000 e K = 15000. A quantidade de iterações máximas dos algoritmos é igual a $IT = 300$.

Todas as fontes U^m, $\forall m \in \{1, 2, 3\}$ desta simulação tem uma probabilidade de erro ρ com a fonte escondida U^0 igual a $\rho = 0.005$. Quer dizer que duas fontes U^m quaisquer tem uma correlação de $corr(U^i, U^j) = 0.9801$, $\forall i \neq j$. Também esta figura mostra o desempenho de um canal não codificado legendado como "Uncoded" e o limite inferior do desempenho do BER para uma matriz **H** de tipo $LDGM$, desenhado com "$\circ$". A Figura 3.9 mostra o desempenho do algoritmo $DPHBF$ usando distintos valores de β para cada E_b/N_0. A melhora máxima no desempenho do algoritmo $DPHBF$ em relação ao algoritmo $PHBF$ é de aproximadamente

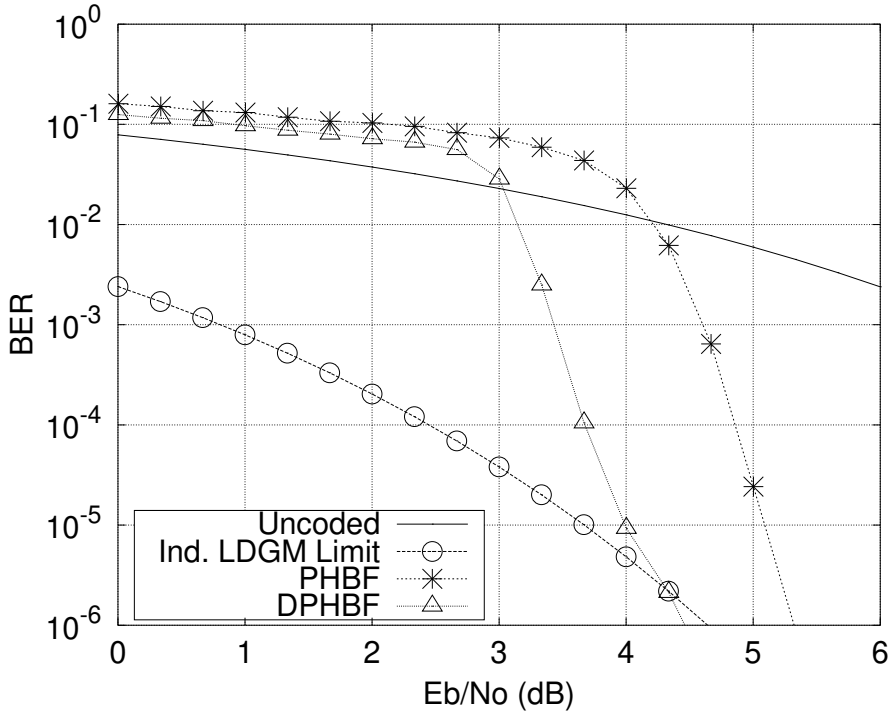

Figura 3.7: BER usando os algoritmos $PHBF$ e $DPHBF$ com uma matriz de tipo $LDGM$, $\mathcal{X} = \{8, 9\}$, K = 2000, N = 3000, $\rho = 0.005$ e M = 3 quando varia β.

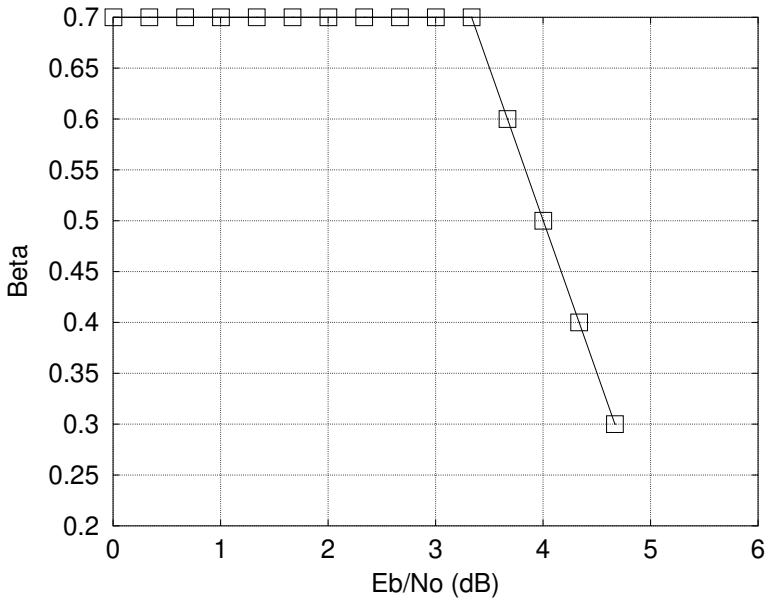

Figura 3.8: Valor de β no desempenho dos algoritmos $PHBF$ e $DPHBF$ com uma matriz de tipo $LDGM$, $\mathcal{X} = \{8, 9\}$, K = 2000, N = 3000, $\rho = 0.005$ e M = 3.

$1.4dB$. Nesta figura é interessante ressaltar que o algoritmo $PHBF$ atinge numa seção um desempenho próximo ao limite inferior do desempenho de matrizes $LDGM$. Nessa seção o algoritmo conjunto acompanha a este limite com uma diferença de aproximadamente $1.0dB$. Isto quer dizer que existe um limite como em (Garcia-Frias & Zhong 2003) para o desempenho de matrizes $LDGM$ para fontes conjuntas. Este limite pode ser previsto observando que o limite virtual de Shannon para fontes conjuntas se diferencia com o limite de Shannon em $1.0dB$ como pode ser visto na Figura 2.13 para o caso em que $\rho = 0.005$.

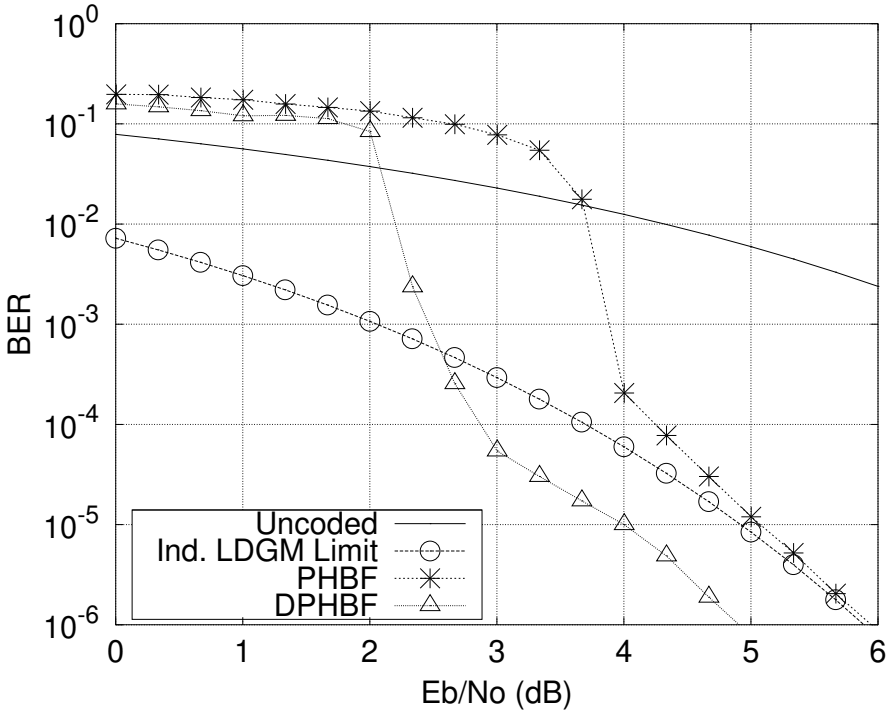

Figura 3.9: BER usando os algoritmos $PHBF$ e $DPHBF$ com uma matriz de tipo $LDGM$, $\mathcal{X} = 9$, K = 15000, N = 30000, $\rho = 0.005$ e M = 3 quando varia β.

Estudo do Desempenho para Diferentes Números de Fontes

A Figura 3.11 mostra o desempenho dos algoritmos $DPHBF$ e $PHBF$ para matrizes **G** e **H** de tipo $LDGM$, a quantidade de uns por coluna de **H** ligados aos bits de informação é de $\mathcal{X} = 5$. O código tem uma taxa r = 2/3, com N = 306 e K = 204. A quantidade de iterações máximas dos algoritmos é igual a $IT = 15$. Todas as fontes U^m, $\forall m \in \{1, 2, ..., \mathrm{M}\}$ desta simulação tem uma probabilidade de erro ρ com a fonte escondida U^0 igual a $\rho = 0.1$, e duas fontes U^m quaisquer têm uma correlação de $corr(U^i, U^j) = 0.64$, $\forall i \neq j$. O desempenho de um canal não codificado é legendado como "Uncoded", o limite inferior do desempenho do BER para uma matriz **H** de tipo $LDGM$ é desenhado com "$\circ$". Todos os algoritmos $DPHBF$ usam um fator de ponderação para a confiabilidade conjunta igual a $\beta = 0.6$. A Figura 3.11 mostra o desempenho do algoritmo $DPHBF$ usando distintos número de fontes, sendo estes valores de M = $\{3, 10, 100\}$. O desempenho do algoritmo $DPHBF$ melhora quando se incrementa o número de fontes, mas como é visto na Figura 2.14 (em relação ao limite virtual de Shannon)

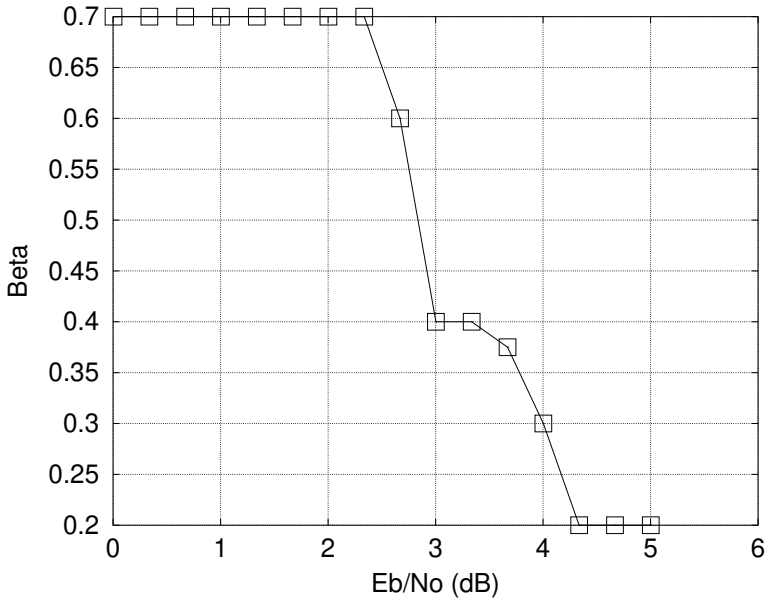

Figura 3.10: Valor de β no desempenho dos algoritmos $PHBF$ e $DPHBF$ com uma matriz de tipo $LDGM$, $\mathcal{X} = 9$, K = 15000, N = 30000, $\rho = 0.005$ e M = 3.

a melhora do desempenho de passar de M = 10 a M = 100 é pouco significativa, a maior parte da melhoria no algoritmo poderá ser vista no uso das primeiras M = 10 fontes. É interessante ressaltar que o algoritmo $DPHBF$ sobrepassa em desempenho ao limite inferior para matrizes $LDGM$, mas na Figura 2.14 se prediz que para um $\rho = 0.1$ a melhoria deveria de ser no melhor dos casos de $1.1dB$ para M = 10, para este caso não se atinge este valor limite. É de se esperar que para valores de N maiores, este desempenho se aproxime a este valor.

Estudo do Desempenho para Diferentes Correlações

As Figuras 3.12 e 3.13 mostram o desempenho dos algoritmos $DPHBF$ e $PHBF$ para matrizes **G** e **H** de tipo $LDGM$, a quantidade de uns por coluna de **H** ligados aos bits de informação é de $\mathcal{X} = 5$. O código tem uma taxa r = 2/3, com N = 306 e K = 204. A quantidade de iterações máximas dos algoritmos é igual a $IT = 15$. As fontes $U^m, \forall m \in \{1, 2, 3, 4, 5\}$, nestas simulações tem uma probabilidade de erro com a fonte escondida U^0 igual a $p_m = \{0.0001, 0.05, 0.10, 0.15, 0.20\}$, equivalente a uma correlação com U^0 como mostra a Tabela 3.1.

corr	U^1	U^2	U^3	U^4	U^5
U^0	0.9998	0.9000	0.8000	0.7000	0.6000

Tabela 3.1: Correlação das 5 fontes das Figura 3.12 e Figura 3.13 com a fonte U^0.

Isto é, as fontes U^m tem uma correlação par a par como é descrita na Tabela 3.2.

O desempenho de um canal não codificado é legendado como "Uncoded", o limite inferior do desempenho do BER para uma matriz **H** de tipo $LDGM$ é desenhado com "$\circ$". O desempenho

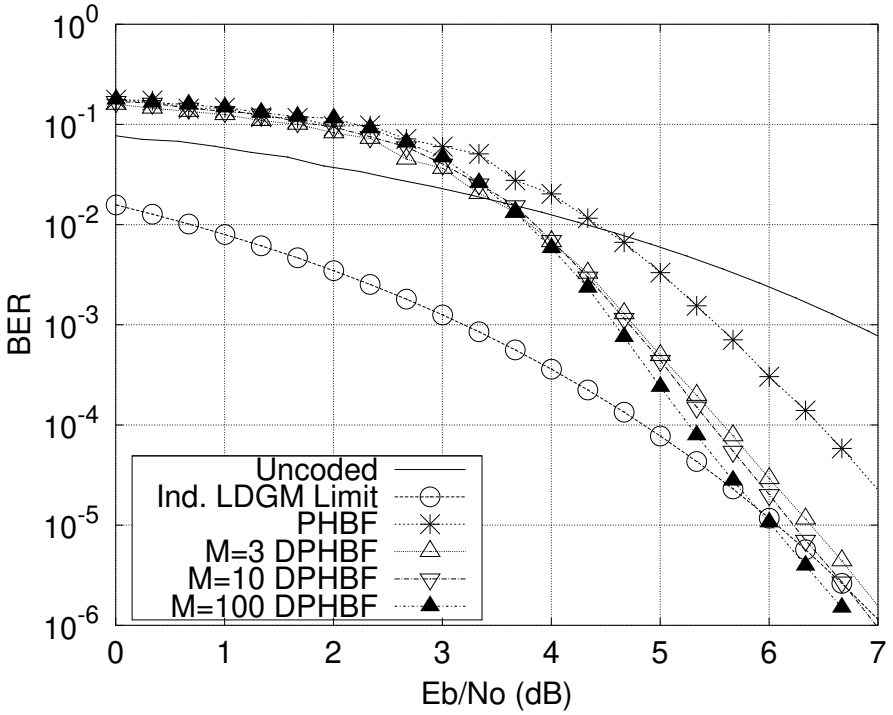

Figura 3.11: BER usando os algoritmos $PHBF$ e $DPHBF$ com uma matriz de tipo $LDGM$, $\mathcal{X} = 5$, K = 204, N = 306, $\rho = 0.1$ e M = $\{3, 10, 100\}$ quando $\beta = 0.6$.

corr	U^1	U^2	U^3	U^4	U^5
U^1	1.00000	0.89982	0.79984	0.69986	0.59988
U^2	0.89982	1.00000	0.72000	0.63000	0.54000
U^3	0.79984	0.72000	1.00000	0.56000	0.48000
U^4	0.69986	0.63000	0.56000	1.00000	0.42000
U^5	0.59988	0.54000	0.48000	0.42000	1.00000

Tabela 3.2: Correlação entre as 5 fontes das Figura 3.12 e Figura 3.13.

do algoritmo $DPHBF$ é desenhado com uma linha pontuada.

Na Figura 3.12 os algoritmos $DPHBF$ usam um fator de ponderação para a confiabilidade conjunta igual a $\beta = 0.6$. A curva $DPHBF$ pertencente a fonte U^1 com um $\rho = 0.0001$ é a de melhor desempenho em contraposição a curva pertencente a fonte U^5 com um $\rho = 0.2$ é a de pior desempenho. Uma curva interessante é a curva pertencente a fonte U^2 com um $\rho = 0.05$, esta curva é ressaltado na figura com $\triangledown$ e tem um comportamento de desempenho não proporcional a correlação, piorando quanto maior é o nível de E_b/N_0. Em geral a curva média de todos os valores $DPHBF$ tem um desempenho melhor que a curva do algoritmo $PHBF$.

Na Figura 3.13 os algoritmos $DPHBF$ usam um fator de ponderação para a confiabilidade conjunta igual a $\beta = 0.8$. O desempenho das curvas $DPHBF$ é semelhante à Figura 3.12 com a diferença que a fonte U^2, ressaltado na figura com $\triangledown$, tem sim um comportamento de desempenho proporcional a correlação, piorando quanto maior é o nível de E_b/N_0. A curva

média de todos os valores $DPHBF$ tem um pior desempenho que a curva do algoritmo $PHBF$ para valores altos de E_b/N_0.

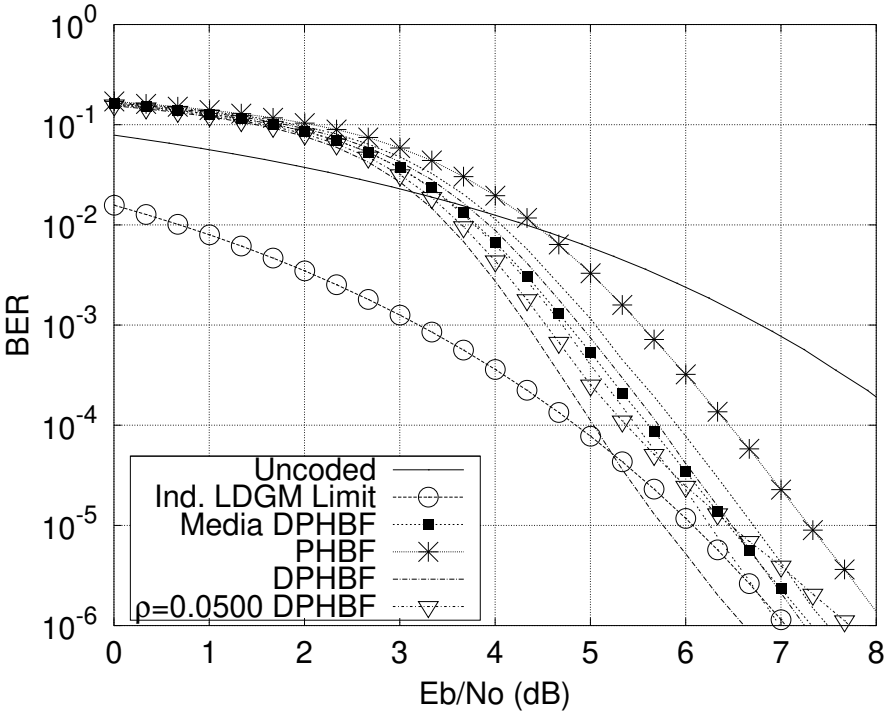

Figura 3.12: BER usando os algoritmos $PHBF$ e $DPHBF$ com uma matriz de tipo $LDGM$, $\mathcal{X} = 5$, K = 204, N = 306, M = 5, $\beta = 0.6$ quando $p_m = \{0.0001, 0.05, 0.10, 0.15, 0.20\}$.

3.4.2 Algoritmo $DPHBF$ com Matrizes $LDPC$

A Figura 3.14 mostra o desempenho dos algoritmos $PHBF$ e $DPHBF$ para uma matriz **H** de tipo $LDPC$ regular e uma matriz **G** de tipo $LDGM$. A quantidade de uns por coluna da matriz **H** é igual a $d_v = 5$. O código tem uma taxa r = 1/2, com N = 5000 e K = 2500. A quantidade de iterações máximas dos algoritmos é igual a $IT = 150$. Todas as fontes U^m, $\forall m \in \{1, 2, 3\}$ tem uma probabilidade de erro ρ com a fonte U^0 igual a $\rho = 0.005$, quer dizer que duas fontes quaisquer tem uma correlação $corr(U^i, U^j) = 0.9801$, $\forall i \neq j$. O desempenho do algoritmo $PHBF$, é desenhado com "$*$", e o desempenho do algoritmo $DPHBF$ com "$\triangle$", esta curva foi criada usando distintos valores de fator de ponderação β da confiabilidade conjunta, sendo os valores que toma $\beta = \{0.6, 0.8\}$ como mostra a Figura 3.15.

3.5 Desempenho dos Algoritmos WBF e $DSPWBF$

A Figura 3.16 mostra o desempenho dos algoritmos WBF e $DSPWBF$ para matrizes **G** e **H** de tipo $LDGM$, a quantidade de uns por coluna de **H** ligados aos bits de informação é de $\mathcal{X} = 5$. O código tem uma taxa r = 2/3, com N = 306 e K = 204. A quantidade de iterações máximas dos algoritmos é igual a $IT = 15$. As fontes $U^m, \forall m \in \{1, 2, 3, 4, 5\}$, destas simulações tem uma

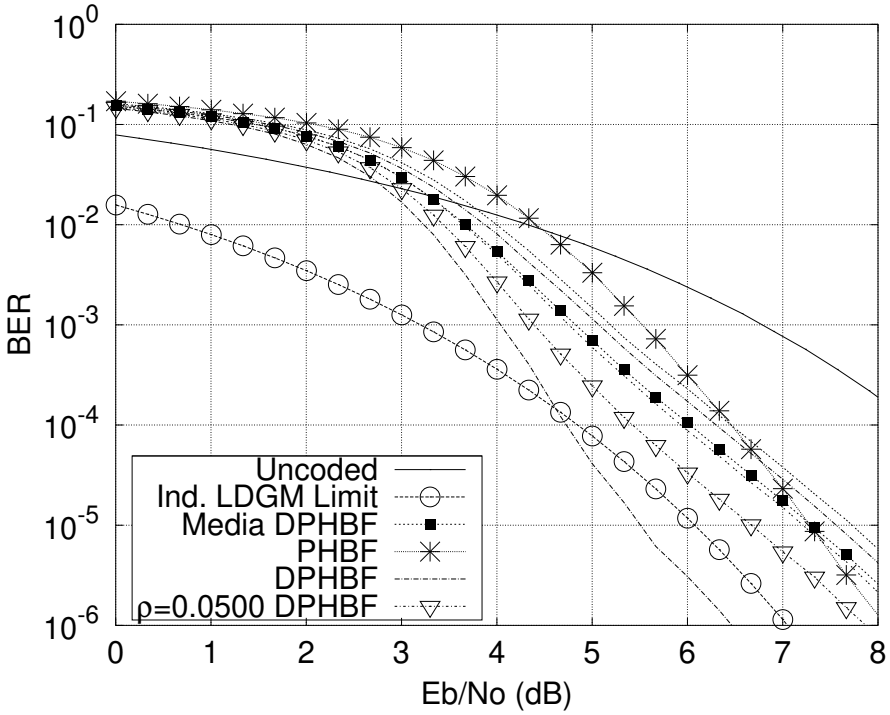

Figura 3.13: BER usando os algoritmos $PHBF$ e $DPHBF$ com uma matriz de tipo $LDGM$, $\mathcal{X} = 5$, K = 204, N = 306, M = 5, $\beta = 0.8$ quando $p_m = \{0.0001, 0.05, 0.10, 0.15, 0.20\}$.

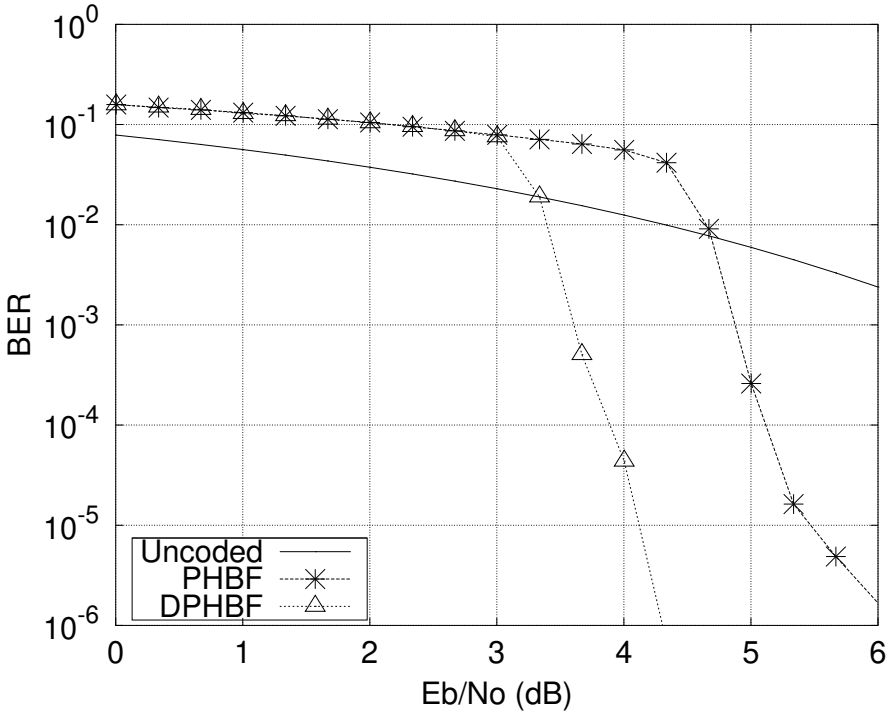

Figura 3.14: BER usando os algoritmos $PHBF$ e $DPHBF$ com uma matriz de tipo $LDGM$, $d_v = 5$, K = 2500, N = 5000, $\rho = 0.005$ e M = 3 quando varia β.

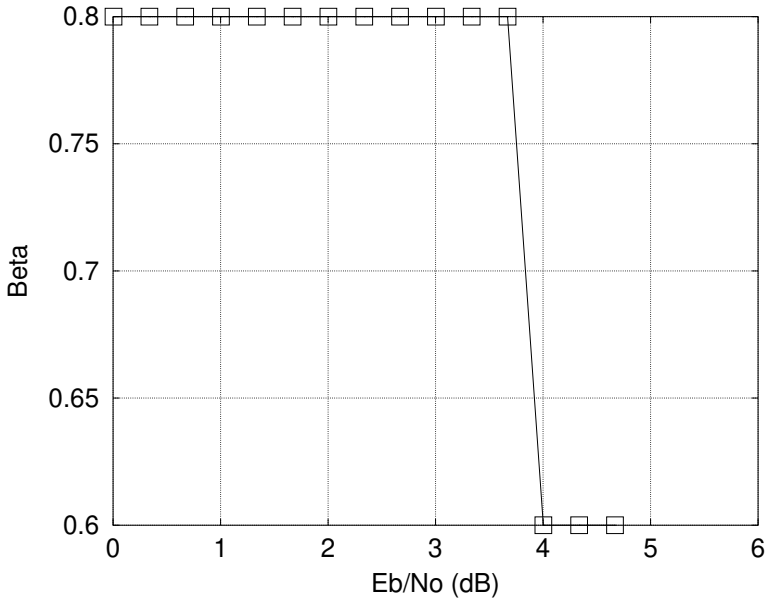

Figura 3.15: Valor de β no desempenho dos algoritmos $PHBF$ e $DPHBF$ com uma matriz de tipo $LDGM$, $d_v = 5$, K = 2500, N = 5000, $\rho = 0.005$ e M = 3.

probabilidade de erro com a fonte escondida U^0 igual a $p_m = \{0.0001, 0.05, 0.10, 0.15, 0.20\}$, equivalente a uma correlação com U^0 como mostra a Tabela 3.1.

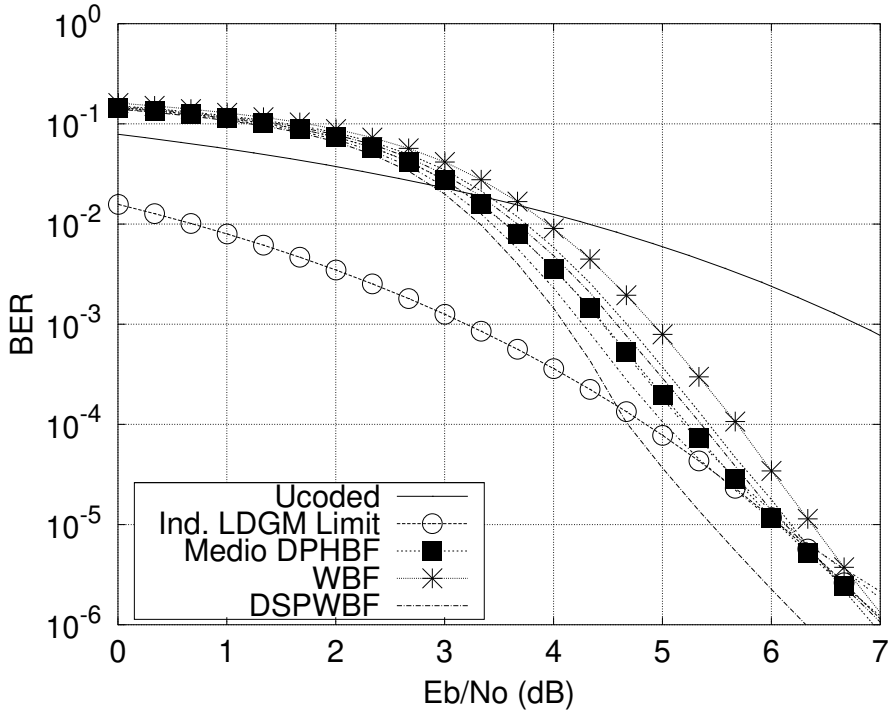

Figura 3.16: BER usando os algoritmos WBF e $DSPWBF$ com uma matriz de tipo $LDGM$, $\mathcal{X} = 5$, K = 204, N = 306, M = 5, $\beta = 1.0$ quando $p_m = \{0.0001, 0.05, 0.10, 0.15, 0.20\}$.

O desempenho de um canal não codificado é legendado como “Uncoded”, o limite inferior do

desempenho do BER para uma matriz **H** de tipo $LDGM$ é desenhado com "$\circ$". O desempenho do algoritmo $DSPWBF$ é desenhado com uma linha pontuada. O algoritmo WBF é desenhado com "$*$". O algoritmo $DSPWBF$ usa um fator de ponderação para a confiabilidade conjunta igual a $\beta = 1.0$. A curva $DSPWBF$ pertencente a fonte U^1 com um $\rho = 0.0001$ é a de melhor desempenho em contraposição a curva pertencente a fonte U^5 com um $\rho = 0.2$ é a de pior desempenho. Em geral a curva média de todos os valores $DSPWBF$ tem um desempenho melhor que a curva do algoritmo WBF.

3.6 Complexidade do Algoritmo Conjunto

Nesta seção analisa-se a complexidade de cálculo do algoritmo conjunto em comparação ao algoritmo independente. É importante ressaltar que mesmo que muitas partes do algoritmo possam ser realizadas em paralelo, o que diminuiria o tempo de decodificação, aqui só será analisado a complexidade das operações sem ter em conta paralelismo.

A Tabela 3.3 mostra a quantidade de operações necessárias para o cálculo do vetor de confiabilidades no algoritmo $DPHBF$ para M fontes correlacionadas, com uma taxa de codificação fonte-canal comum r = K/N e $Iter$ iterações. A complexidade do cálculo da confiabilidade independente (I^m), conjunta (E^m) e total (${E_T}^m$), é calculada para uma fonte e M fontes. B representa as operações do cálculo binário de $S^m = Z^m H^t$ e o cálculo inteiro de $I^m = (2S^m - 1)H$. D representa as operações do cálculo inteiro de $E^m = (\sum_1^{\mathrm{M}-1} I^m)/(\mathrm{M}-1)$. Uma decodificação

Fontes	I^m	E^m	${E_T}^m$
1	B*Iter	D*Iter	(B+D)*Iter
M	B*Iter*M	D*Iter*M	(B+D)*Iter*M

Tabela 3.3: Complexidade do cálculo do vetor de confiabilidades independentes, conjuntas e totais.

independente gasta em total $B * Iter * M$ operações e a decodificação conjunta gasta em total $(B + D) * Iter * M$ operações. Pode-se ver que B cresce em proporção ao crescimento da quantidade de bits codificados N, enquanto que D cresce linearmente em proporção ao crescimento da quantidade M de fontes envolvidas na decodificação. Assim o algoritmo de decodificação conjunta tem uma complexidade tratável para um número moderado de fontes.

3.7 Observações Sobre a Decodificação Conjunta

- O algoritmo $PHBF$ distribuído ($DPHBF$) tem um melhor desempenho quando comparado ao algoritmo $PHBF$, como pode ser visto nas Figuras 3.5, 3.7 e 3.9. Este melhor desempenho depende do parâmetro β de ponderação entre as confiabilidades independentes e conjuntas. É melhor sempre otimizar o parâmetro β para cada valor de nível de ruído E_b/N_0. Valores de β altos terão um bom desempenho para valores de E_b/N_0 baixo e com valores de β baixos se obterá um melhor desempenho em valores de E_b/N_0 altos.

- Dado um conjunto de fontes correlacionadas, a fonte que terá o melhor desempenho no algoritmo $DPHBF$ é a fonte com a maior correlação com a fonte U^0, isto pode ser visto nas Figuras 3.12 e 3.13.

- Em todas as decodificações apresentadas se usa uma matriz **G** de tipo $LDGM$, mesmo quando a matriz **H** é de tipo $LDPC$. Isto é para conservar a correlação entre fontes correlacionadas codificadas, pelo menos na parte sistemática do vetor codificado.

- Quando se usa um código $LDPC$ o fator de ponderação β não pode ser aplicado a todo o vetor de confiabilidades conjuntas. Isto é porque a correlação só existe nos bits de informação. Assim existe confiabilidade conjunta só nos bits de informação. Isto não é um problema em matrizes **H** de tipo $LGDM$ porque o peso das confiabilidades é muito baixo nos bits de paridade quanto comparados aos bits de informação, pelo qual β pode ser usado em todo o vetor sem mostrar variação no desempenho do algoritmo conjunto.

- Quando se usa um algoritmo conjunto como o $DPHBF$ e ele não mostra um melhor desempenho quando comparado ao algoritmo independente, só um desempenho igual quando se usa $\beta = 0$, significa que o algoritmo necessita mais iterações. Isto é porque o algoritmo conjunto ressalta as confiabilidades dos bits errados provocando que a correção dos erros se faça bit a bit na maioria das vesses, comportando-se como um algoritmo serial. Uma boa ideia nesses casos é predizer a quantidade de iterações IT como $IT = NQ(\sqrt{2rE_b/N_0})$, sendo E_b/N_0 o valor em que se deseja que aconteça a queda da curva de desempenho do algoritmo conjunto, $r = \mathrm{K/N}$ sendo a taxa de codificação e N o número de bits na palavra codificada.

- O algoritmo $SPWBF$ distribuído ($DSPWBF$) tem um melhor desempenho quando compa-
  rado ao algoritmo WBF, como pode ser visto na Figura 3.12. Este melhor desempenho depende do parâmetro β de ponderação entre as confiabilidades independentes e conjuntas.

Capítulo 4

Decodificação Conjunta Aplicada ao Problema CEO

Este capítulo tem como contribuição a definição de uma regra de decisão para a predição da fonte U^0 no problema CEO descrito na Figura 4.1. Esta figura mostra o modelo para a transmissão de dados, onde M fontes correlacionadas são geradas a partir de uma fonte comum U^0. A informação das fontes sofre uma codificação de fonte e de canal. Dado que não existe comunicação entre os codificadores estes trabalham de forma independente. A informação codificada é enviada por um canal $BI - AWGN$. A informação recebida na saída dos canais é decodificada de forma conjunta para obter $\hat{U}^m$, $m \in \{1, 2, ..., M\}$, que são versões aproximadas das fontes U^m.

Todas as fontes $\hat{U}^m$ tem uma parte da informação de U^0, de modo que é possível obter $\hat{U}^0$, uma versão aproximada de U^0 a partir das fontes $\hat{U}^m$. A este cenário denomina-se problema CEO (do inglês, Chief Executive Officer). Este problema define a dificuldade de um diretor executivo em obter a informação da fonte U^0, quando ele apenas conhece a informação distorcida fornecida pelas fontes U^m.

4.1 Decodificação CEO

A Figura 4.2 representa um modelo simplificado da Figura 4.1, útil para representar a geração de fontes binárias correlacionadas na analise do problema CEO.

O modelo de geração de fontes correlacionadas da Figura 4.2 é equivalente ao descrito na Figura 1.10. O somador XOR é substituído por um canal BSC com probabilidade de erro $P(e^i = 1) = p_i$.

O problema CEO necessita de uma regra de decisão para obter o valor de $\hat{U}^0$, que é uma estimação de U^0, conhecendo sómente os valores de $U^1, U^2, U^3, ...$ e U^{M}.

É interessante perguntar-se que valor tem $H(U^0|U^1 \dots U^{\mathrm{M}})$. Esta expressão indica a informação que se tem de U^0 dado que são conhecidos os valores de U^m, $\forall m \in \{1, 2, ..., \mathrm{M}\}$. A informação fornecida por $H(U^0|U^1 \dots U^{\mathrm{M}})$ equivale ao desconhecimento que se tem de U^0. Ou dito de uma outra maneira, "a incerteza de U^0 dado que se conhece os valores de U^m". Por exemplo dado o caso hipotético que seja possível calcular analiticamente U^0 através do conhecimento

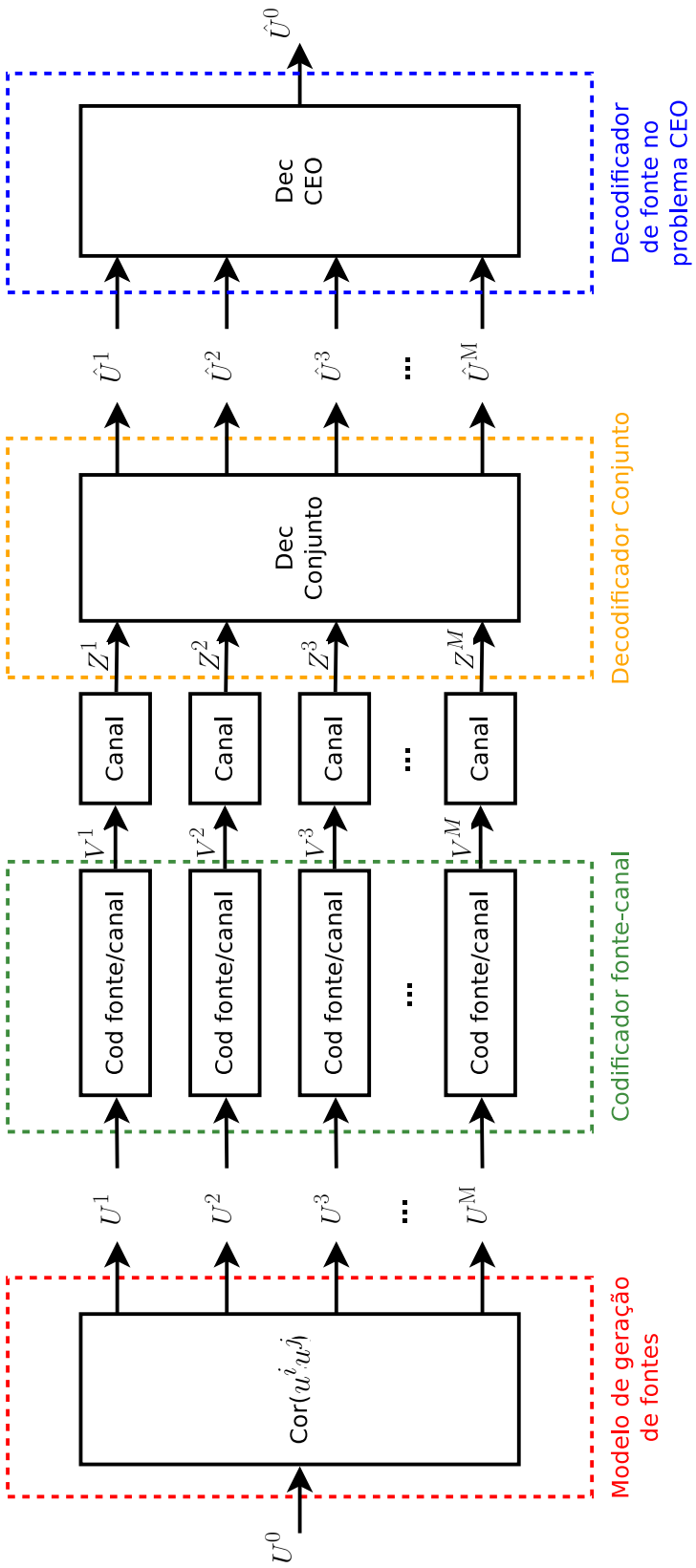

Figura 4.1: Modelo para a transmissão de dados num problema CEO binário

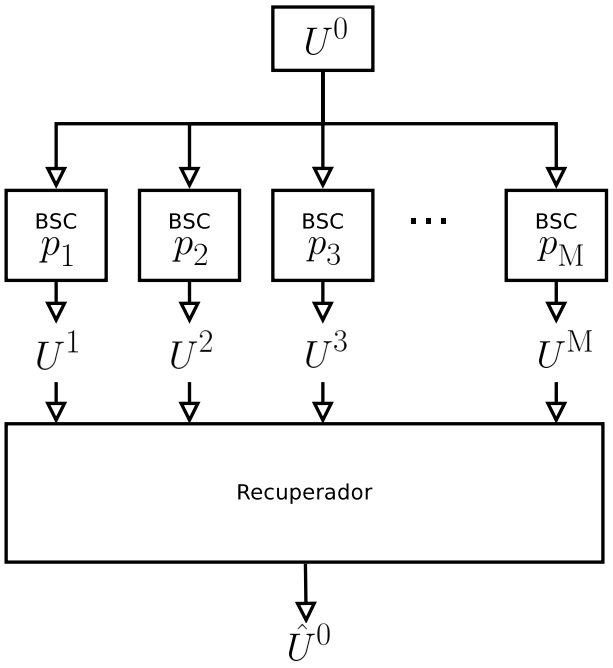

Figura 4.2: Gráfico da geração de fontes correlacionadas para o problema CEO.

de $U^1 \dots U^{\mathrm{M}}$, se obteria $H(U^0|U^1 \dots U^{\mathrm{M}}) = H(U^0|U^0) = 0$. O que mostra que esta expressão não fornece a informação recuperável de U^0, mas sim a informação perdida. Define-se esta informação perdida como

$$h\left(P(u^0 \neq \hat{u}^0)\right) = H(U^0|U^1 \dots U^{\mathrm{M}}), \tag{4.1}$$

onde $h(.)$ é a função entropia binária e $P(u^0 \neq \hat{u}^0)$ é a probabilidade de que u^0 seja diferente de $\hat{u}^0$. Da equação (4.1) pode-se obter uma expressão limite para a probabilidade de erro na estimação da fonte U^0

$$P(u^0 \neq \hat{u}^0) = h^{-1}\left(H(U^0|U^1 \dots U^{\mathrm{M}})\right). \tag{4.2}$$

Para conhecer o valor de $H(U^0|U^1 \dots U^{\mathrm{M}})$ segue-se o procedimento apresentado no Apêndice A.14.

4.2 Decodificação da Fonte sem o Uso do Ruído de Canal

Uma boa ideia para obter a estimação de U^0 no modelo da Figura 4.2 é usar o critério MAP (Máximo a posteriori) fazendo o quociente

$$\phi = \frac{P(u^0 = 1|U^1U^2U^3 \dots U^{\mathrm{M}})}{P(u^0 = 0|U^1U^2U^3 \dots U^{\mathrm{M}})}, \tag{4.3}$$

para obter a aproximação $\hat{U}^0$ como

$$\hat{u}^0 = \begin{cases} 1 & \text{if } \phi > 1 \\ 0 & \text{if } \phi \leq 1 \end{cases}, \tag{4.4}$$

Seguindo o procedimento do Apêndice A.9 a equação (4.3) fica expressada como segue

$$\phi = \frac{P(U^1|u^0=1)P(U^2|u^0=1)\dots P(U^{\mathrm{M}}|u^0=1)}{P(U^1|u^0=0)P(U^2|u^0=0)\dots P(U^{\mathrm{M}}|u^0=0)}\frac{P(u^0=1)}{P(u^0=0)}. \tag{4.5}$$

Um procedimento similar usando MAP é descrito em (Ferrari et al. 2012) e (Ferrari et al. 2014), A diferença com este artigo, é que este usa LLR (do inglês, Log Likelihood Ratio) em vez de probabilidades de erro de canal.

Exemplo 4.2.1 (Probabilidade p_i constante) *Para o caso especifico em que $p_i = \rho$ e $P(u^0 = 1) = 0.5$, a equação (4.5) pode ser simplificada como*

$$\phi = \left(\frac{1-\rho}{\rho}\right)^{\eta}\left(\frac{\rho}{1-\rho}\right)^{\mathrm{M}-\eta} = \left(\frac{1-\rho}{\rho}\right)^{2\eta-\mathrm{M}}, \tag{4.6}$$

onde η é o número de vezes em que $u^m = 1$. Aplicando o logaritmo natural a

$$\phi > 1 \tag{4.7}$$

obtemos

$$(2\eta - \mathrm{M})ln\left(\frac{1-\rho}{\rho}\right) > 0, \tag{4.8}$$

que implica que

$$\eta > \mathrm{M}/2 \quad se \quad \rho < 0.5 \tag{4.9}$$

ou

$$\eta < \mathrm{M}/2 \quad se \quad \rho > 0.5. \tag{4.10}$$

Simplificando-se a equação (4.4) no caso em que $\rho < 0.5$, como

$$\hat{u}^0 = \begin{cases} 1 & if \ \ \eta > \mathrm{M}/2 \\ 0 & if \ \ \eta \leq \mathrm{M}/2 \end{cases}. \tag{4.11}$$

E no caso em que $\rho > 0.5$ a equação (4.4) simplifica-se como

$$\hat{u}^0 = \begin{cases} 1 & if \ \ \eta < \mathrm{M}/2 \\ 0 & if \ \ \eta \geq \mathrm{M}/2 \end{cases}. \tag{4.12}$$

Na Figura 4.3 pode-se ver uma simulação do desempenho ao realizar a estimação da fonte U^0 para distintos valores M *do número de fontes correlacionadas, usando a equação (4.11) para distintos valores de ρ.*

No artigo (Haghighat et al. 2008) apresenta-se uma equação que expressa o desempenho na predição de $\hat{U}^0$ no Exemplo 4.2.1. Esta equação é

$$P(\hat{u}^0 \neq u^0) = \begin{cases} \frac{1}{2}\binom{\mathrm{M}}{\frac{\mathrm{M}}{2}}\rho^{\mathrm{M}/2}(1-\rho)^{\mathrm{M}/2} + \sum_{m=\frac{\mathrm{M}}{2}+1}^{\mathrm{M}}\binom{\mathrm{M}}{m}\rho^m(1-\rho)^{\mathrm{M}-m} & ,\mathrm{M} = par \\ \sum_{m=\frac{\mathrm{M}+1}{2}}^{\mathrm{M}}\binom{\mathrm{M}}{m}\rho^m(1-\rho)^{\mathrm{M}-m} & ,\mathrm{M} = impar \end{cases}. \tag{4.13}$$

Pode-se deduzir da equação anterior que:

Proposição 4.1 *Para um $\rho \neq 0$, $P(\hat{u}^0 \neq u^0)$ nunca tomará o valor zero. Isto quer dizer que nunca será possível obter toda a informação da fonte U^0, mas $P(\hat{u}^0 \neq u^0)$ pode tomar valores próximos a zero quando ρ tende a zero ou* M *tende a infinito.*

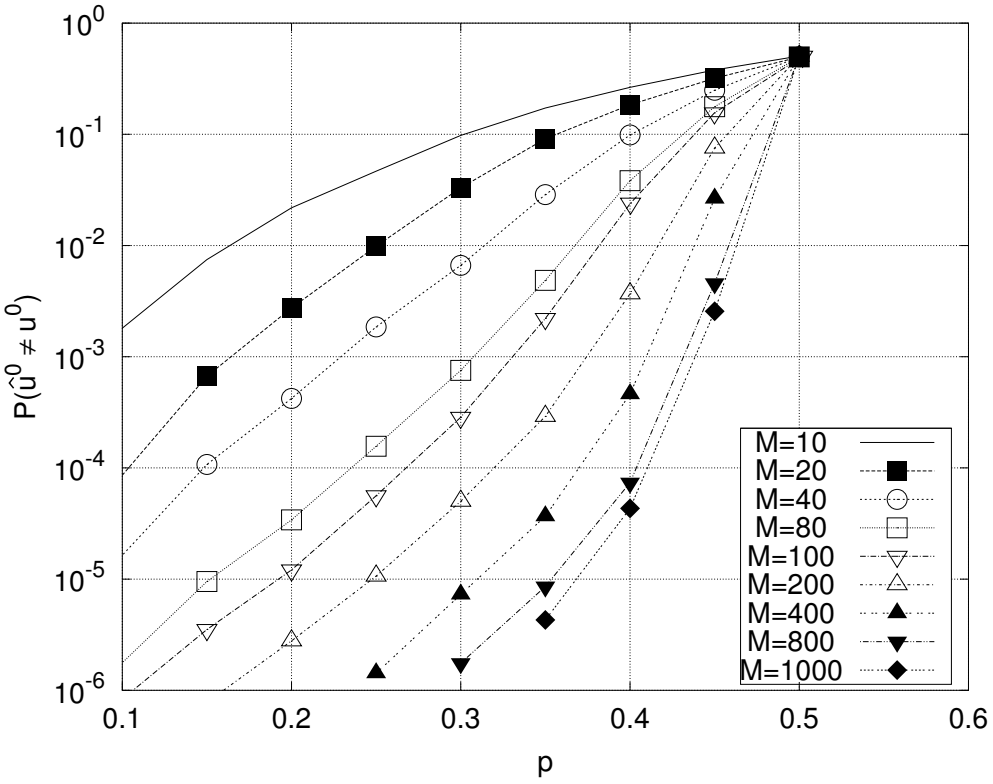

Figura 4.3: Gráfico do erro de bit na predição de U^0 para M fontes correlacionadas, com M $= \{10, 20, 40, 80, 100, 200, 400, 800, 1000\}$ e $P(u^0 = 1) = 0.5$.

4.2.1 Estudo do Desempenho do Problema CEO sem o Uso do Ruído de Canal para Decodificação $PHBF$ e $DPHBF$

As Figuras 4.4 e 4.5 mostram o desempenho dos algoritmos para obter a predição $\hat{U}^0$ de U^0, desprezando a probabilidade de erro da decodificação do canal em comparação à probabilidade de erro do modelo de geração de fontes. A Figura 4.4 usa para sua execução os dados obtidos na simulação da Figura 3.12. A Figura 4.5 usa para sua execução os dados obtidos na simulação da Figura 3.13. Em todas as simulações foi usado o modelo de sistema como mostrado na Figura 4.1, seguindo este modelo, a curva denotada como "CEO" representa o desempenho do algoritmo para a decodificação CEO proposta na Seção 4.2 usando diretamente os dados das fontes U^m, $\forall m \in \{1, 2, ..., 5\}$ sem ruído de canal. A curva denotada por "*Uncoded* CEO" representa o desempenho do algoritmo para decodificação CEO proposta na Seção 4.2 usando diretamente os dados das fontes Z^m com ruído de canal e sem nenhum tipo de codificação. A curva denotada por "$PHBF$ CEO" representa o desempenho do algoritmo para a decodificação CEO proposta na Seção 4.2 usando os dados das fontes $\hat{U}^m$ após a decodificação $PHBF$. A curva denotada por "$DPHBF$ CEO" representa o desempenho do algoritmo para a decodificação CEO proposta na Seção 4.2 usando os dados das fontes $\hat{U}^m$ após a decodificação $DPHBF$. As curvas "*Uncoded* CEO", "$PHBF$ CEO" e "$DPHBF$ CEO" dependem do nível de ruído. Como pode-se ver nas figuras, o desempenho do algoritmo CEO após aplicar o algoritmo $DPHBF$ na decodificação do canal é superior ao desempenho do algoritmo CEO após aplicar o algoritmo $PHBF$. E ambas as curvas só são piores em termos de desempenho, quando comparados ao algoritmo CEO de

dados não codificados, para valores baixos da relação E_b/N_0.

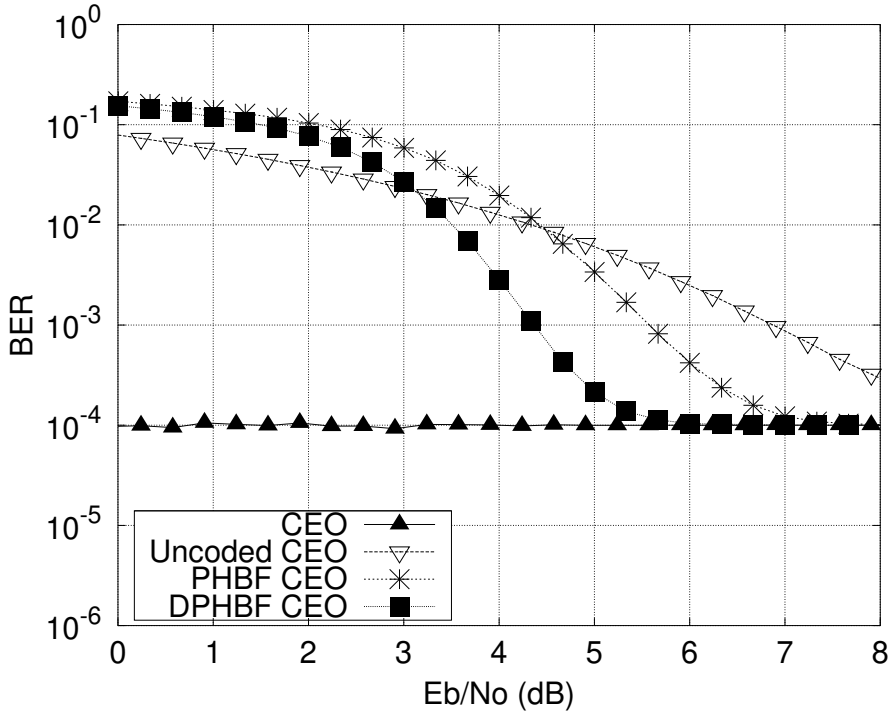

Figura 4.4: BER na estimação de U^0 no problema CEO em uma matriz de tipo $LDGM$, $\mathcal{X} = 5$, K = 204, N = 306, M = 5, $\beta = 0.6$ quando $p_m = \{0.0001, 0.05, 0.10, 0.15, 0.20\}$.

4.2.2 Estudo do Desempenho do Problema CEO sem o Uso do Ruído de Canal para Decodificação WBF e $DSPWBF$

A Figura 4.6 mostra o desempenho do algoritmo para obter a predição $\hat{U}^0$ de U^0 para distintos casos. A Figura 4.6 usa para sua execução os dados obtidos na simulação da Figura 3.16. Em todas as simulações foi usado o modelo de sistema como mostrado na Figura 4.1. Seguindo este modelo, a curva denotada por "CEO" representa o desempenho do algoritmo para a decodificação CEO proposto na Seção 4.2 usando diretamente os dados das fontes U^m, $\forall m \in \{1, 2, ..., 5\}$ sem ruído. A curva denotada por "*Uncoded* CEO" representa o desempenho do algoritmo para a decodificação CEO proposta na Seção 4.2 usando diretamente os dados das fontes Z^m com ruído e sem nenhum tipo de codificação. A curva denotada por "WBF CEO" representa o desempenho do algoritmo para a decodificação CEO proposta na Seção 4.2 usando os dados das fontes $\hat{U}^m$ após a decodificação WBF. A curva denotada por "$DSPWBF$ CEO" representa o desempenho do algoritmo para a decodificação CEO proposta na Seção 4.2 usando os dados das fontes $\hat{U}^m$ após a decodificação $DSPWBF$. Como pode-se ver nas figura, o desempenho do algoritmo CEO após aplicar o algoritmo $DSPWBF$ na decodificação de canal é superior ao desempenho do algoritmo CEO após aplicar o algoritmo WBF. E ambas as curvas só são melhores em termos de desempenho, quando comparados ao algoritmo CEO de dados não codificados, para valores altos de relação E_b/N_0.

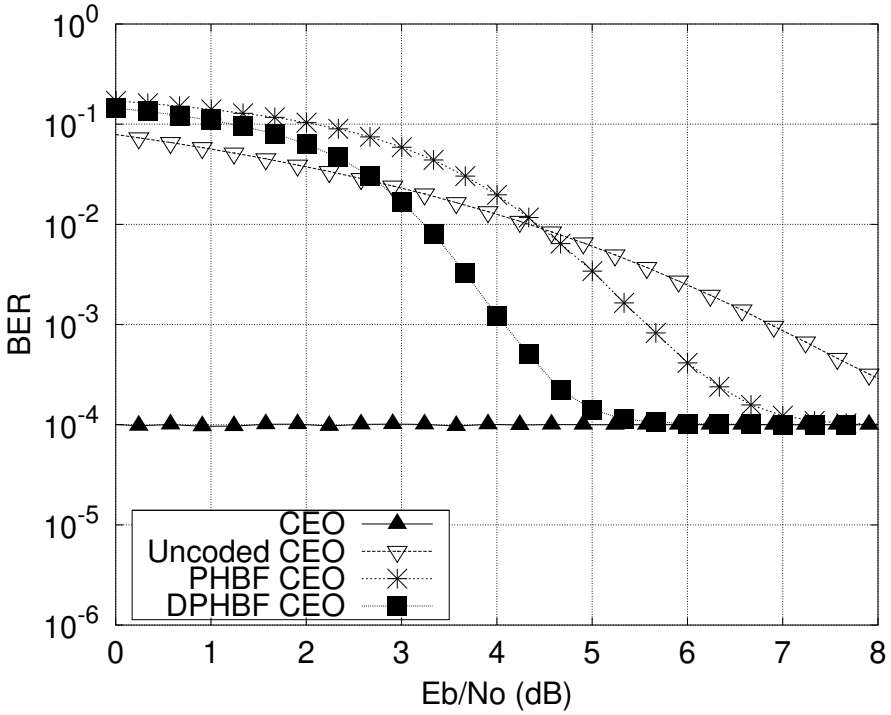

Figura 4.5: BER na estimação de U^0 no problema CEO em uma matriz de tipo $LDGM$, $\mathcal{X} = 5$, K = 204, N = 306, M = 5, $\beta = 0.8$ quando $p_m = \{0.0001, 0.05, 0.10, 0.15, 0.20\}$.

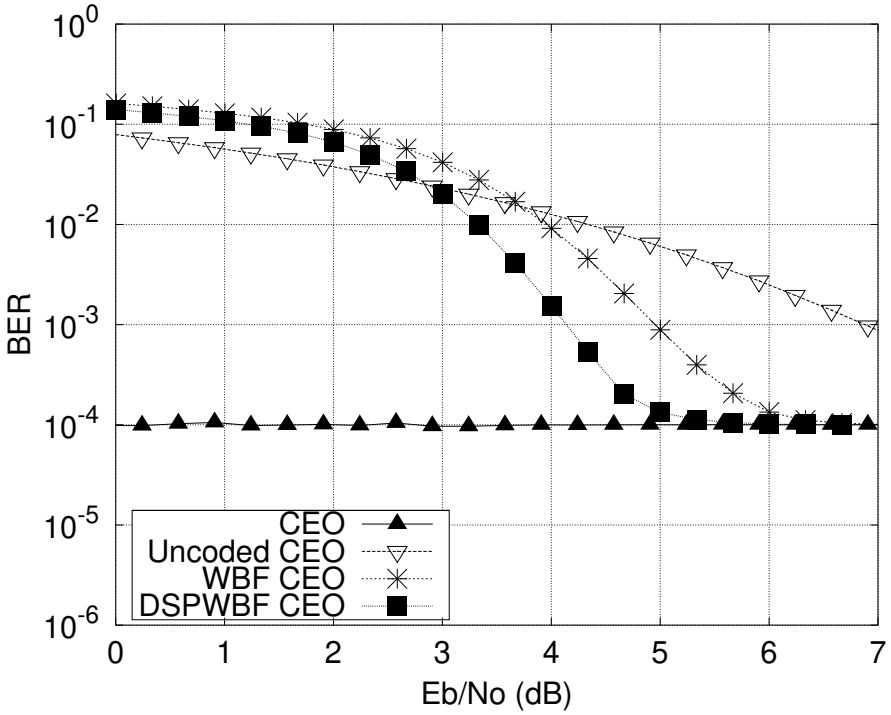

Figura 4.6: BER na estimação de U^0 no problema CEO com uma matriz de tipo $LDGM$, $\mathcal{X} = 5$, K = 204, N = 306, M = 5, $\beta = 0.6$ quando $p_m = \{0.0001, 0.05, 0.10, 0.15, 0.20\}$.

4.3 Decodificação de Fonte Usando o Ruído de Canal

Se é desejado resolver um problema de tipo CEO, o valor de $\hat{U}^0$ fica expressado pela equação (4.4). Mas se é necessário calcular $\hat{U}^0$, quando as fontes U^m, $\forall m \in \{1, 2, ..., \mathrm{M}\}$ passam por canais ruidosos, se terá que achar o canal equivalente de tipo BSC com probabilidade de erro p_{c_m}, como mostra a Figura 4.7. Estas probabilidades p_{c_m} representam a probabilidades de erro de $\hat{U}^m$ ao predizer U^m após a decodificação fonte-canal, estas probabilidades dependem do nível de ruído E_b/N_0. Estes valores são obtidos na decodificação estudada no Capítulo 3.

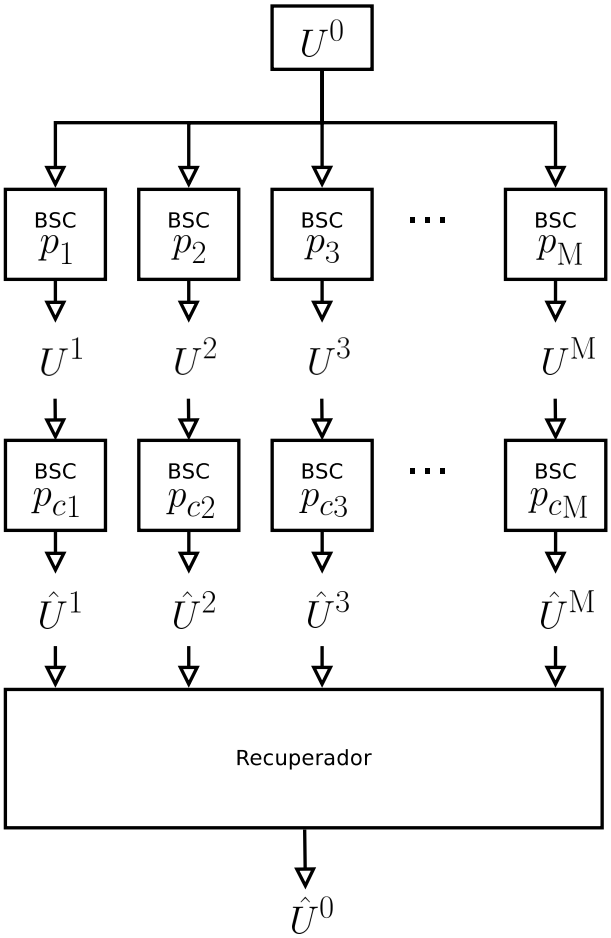

Figura 4.7: Gráfico da geração de fontes correlacionadas para o problema CEO com ruído.

Obtidos os valores de p_{c_m}, a fonte U^0 passará por dois canais BSC em serie e estes canais podem ser trocados por um só canal BSC. O modelo com um canal BSC equivalente é mostrado na Figura 4.8.

O canal equivalente a estes dois canais em serie é um novo canal BSC com probabilidade de erro

$$p_{b_m} = p_m + p_{c_m} - 2p_m p_{c_m}. \tag{4.14}$$

Para estes novos canais o valor de decisão ϕ da equação (4.5) fica reformulada como segue

$$\phi_{ruido} = \frac{P(\hat{U}^1|u^0=1)P(\hat{U}^2|u^0=1)\dots P(\hat{U}^m|u^0=1)}{P(\hat{U}^1|u^0=0)P(\hat{U}^2|u^0=0)\dots P(\hat{U}^m|u^0=0)} \frac{P(u^0=1)}{P(u^0=0)}, \tag{4.15}$$

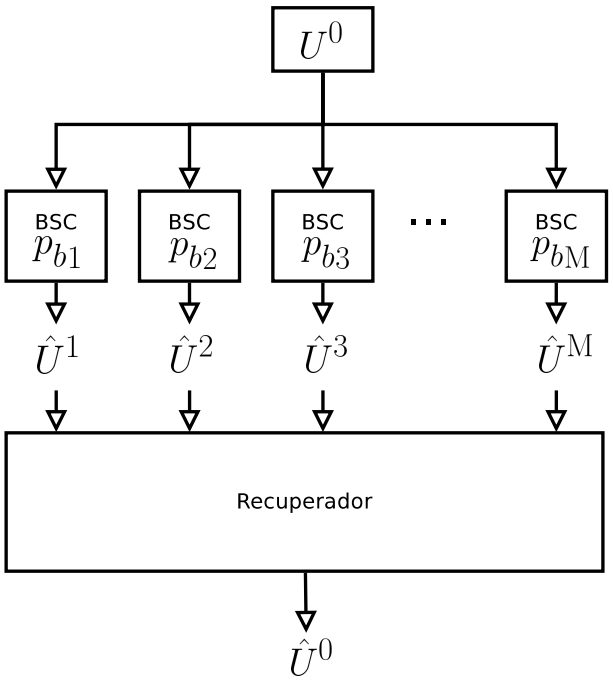

Figura 4.8: Gráficos equivalentes da geração de fontes correlacionadas para o problema CEO com ruído.

$$P(\hat{U}^m|u^0=0)=\begin{cases}p_{b_m} & \text{if } \hat{u}^m=1\\ 1-p_{b_m} & \text{if } \hat{u}^m=0\end{cases}, \tag{4.16}$$

$$P(\hat{U}^m|u^0=1)=\begin{cases}1-p_{b_m} & \text{if } \hat{u}^m=1\\ p_{b_m} & \text{if } \hat{u}^m=0\end{cases}. \tag{4.17}$$

Reformulando a equação (4.15) para obter uma expressão mais ordenada, se obtém:

$$\phi_{ruido}=t_1t_2\ldots t_m\frac{P(u^0=1)}{P(u^0=0)}, \tag{4.18}$$

$$t_i=\begin{cases}\frac{1-p_{b_m}}{p_{b_m}} & \text{if } \hat{u}^m=1\\ \frac{p_{b_m}}{1-p_{b_m}} & \text{if } \hat{u}^m=0\end{cases}. \tag{4.19}$$

4.3.1 Estudo do Desempenho do Problema CEO com Ruído de Canal para Decodificação $PHBF$ e $DPHBF$

A Figura 4.9 mostra o desempenho dos algoritmos para obter a predição $\hat{U}^0$ de U^0, usando a probabilidade de erro da decodificação do canal como uma probabilidade de erro de um canal BSC em cascata com o modelo de geração de fontes. A Figura 4.9 usa para sua execução os dados obtidos na simulação da Figura 3.12. Em todas as simulações foi usado o modelo de sistema como mostrado na Figura 4.1, seguindo este modelo, a curva denotada por "CEO" representa o desempenho do algoritmo para a decodificação CEO proposta na Seção 4.2 usando diretamente os dados das fontes U^m, $\forall m \in \{1,2,...,5\}$ sem ruído. A curva denotada por "*Uncoded* CEO" representa o desempenho do algoritmo para a decodificação CEO proposta na

Seção 4.3 usando diretamente os dados das fontes Z^m com ruído de canal e sem nenhum tipo de codificação. A curva denotada por "$PHBF\ CEO$" representa o desempenho do algoritmo para a decodificação CEO proposta na Seção 4.3 usando os dados das fontes $\hat{U}^m$ após a decodificação $PHBF$. A curva denotada por "$DPHBF\ CEO$" representa o desempenho do algoritmo para a decodificação CEO proposta na Seção 4.3 usando os dados das fontes $\hat{U}^m$ após a decodificação $DPHBF$. As curvas "$Uncoded\ CEO$", "$PHBF\ CEO$" e "$DPHBF\ CEO$" apresentam uma

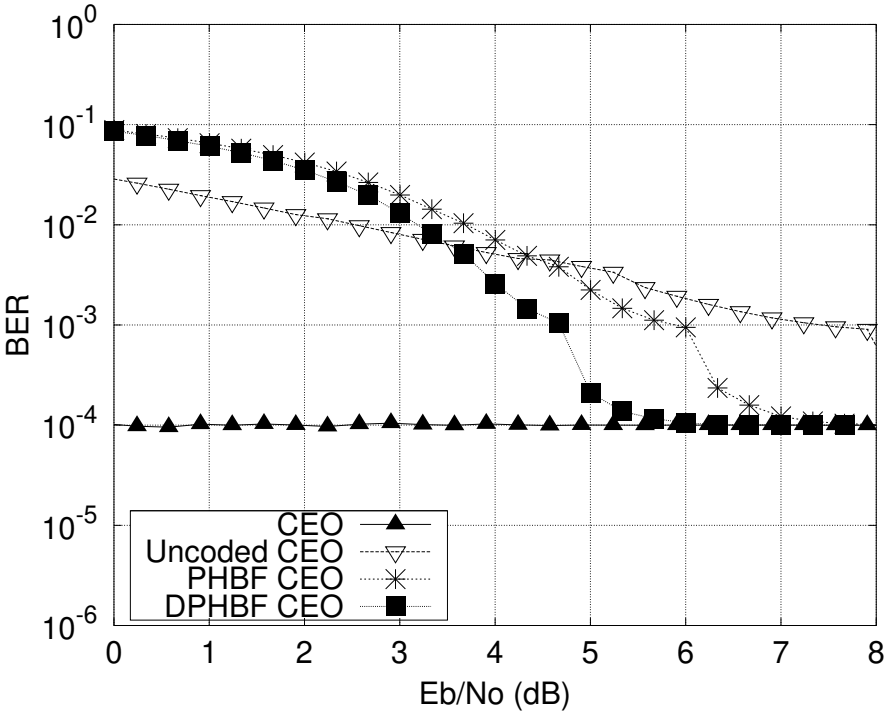

Figura 4.9: BER na estimação de U^0 no problema CEO com uma matriz de tipo $LDGM$, $\mathcal{X} = 5$, K = 204, N = 306, M = 5, $\beta = 0.6$ quando $p_m = \{0.0001, 0.05, 0.10, 0.15, 0.20\}$.

ligeira descontinuidade próxima a um valor de $BER = 10^{-3}$. Esta descontinuidade é produto da distribuição irregular das correlações das fontes. Como pode-se ver na Tabela 3.1 a probabilidade de erro de bit entre a fonte U^1 e a fonte U^0 é de 10^{-4}. Pelo fato de que esta fonte sobressai grandemente em correlação com U^0, esta domina a decodificação CEO. Assim, seguindo o modelo da Figura 4.7 a descontinuidade acontece quando a probabilidade de erro de bit na decodificação do canal p_{c1} do canal U^1 fica próxima em valor a p_1, como pode ser visto na Figura 4.10.

Comparando os gráficos das Figuras 4.4 e 4.9 observa-se que usar o modelo de decodificação CEO da Seção 4.3 tem um melhor desempenho que o modelo de decodificação CEO da Seção 4.2, exceto para valores de p_{c1} e p_1 próximos. Esta exceção acontece no caso especial em que a correlação da fonte U^1 é superior às outras, o que implica que alguma aproximação ruim de p_{c1} gere descontinuidades apreciáveis na decodificação CEO.

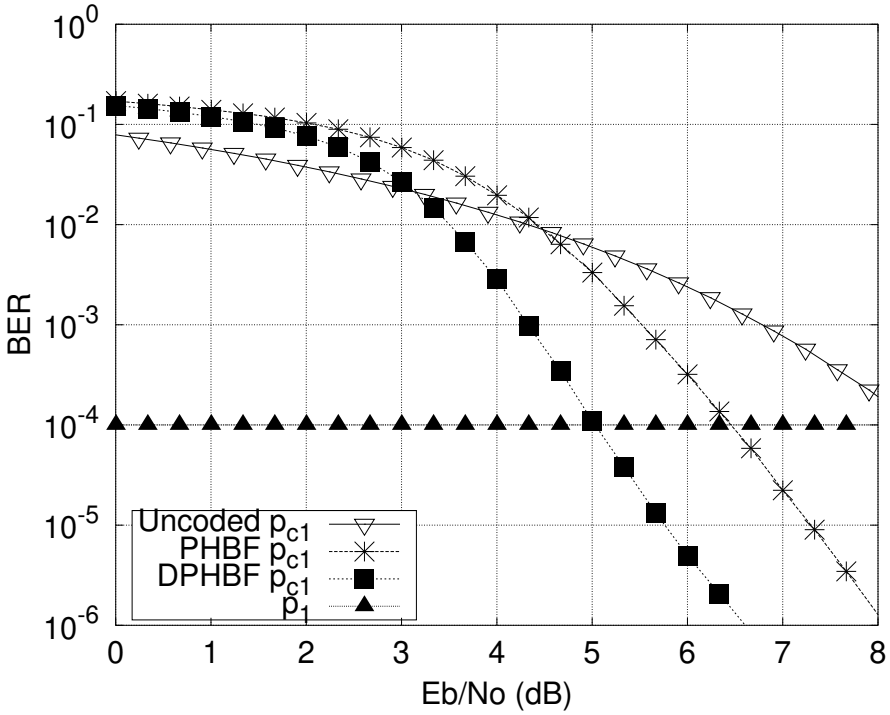

Figura 4.10: Valores de p_1 e p_{c1} para canais não codificados, com decodificação $PHBF$ e decodificação $DPHBF$.

4.4 Observações Sobre a Decodificação CEO

- O desempenho na decodificação CEO está limitado pelo desempenho da fonte com a maior correlação, como pode ser visto nas Figuras 4.4 e 4.5. Nelas o desempenho médio no seu algoritmo $DPHBF$ é pior no sistema da Figura 4.5, mas a decodificação CEO tem um melhor desempenho porque a fonte com mais correlação com U^0 tem melhor desempenho, mesmo que a média seja pior.

- Na decodificação CEO se a fonte com a maior correlação tem uma correlação próxima a 1.0, significa que a probabilidade de erro p_i com a fonte escondida U^0 é próxima a zero, então essa probabilidade p_i define o valor minimo da probabilidade de erro $P(u^0 \neq \hat{u}^0)$. Como pode ser visto nas Figuras 4.4 e 4.5, onde $p_i = 0.0001$ e o valor mínimo de $P(u^0 \neq \hat{u}^0)$ é próximo a 0.0001.

Capítulo 5

Conclusões

No Capítulo 2 desta tese apresenta-se uma taxa ótima para uma codificação conjunta fonte-canal que pode ser calculada para um número moderado de fontes. Em caso que se deseje achar uma taxa de codificação comum r a complexidade do cálculo cresce exponencialmente, sendo necessário fazer $2^{\mathrm{M}} - 1$ comparações para obter a taxa de codificação. No caso da taxa soma máxima o comprimento da inequação matricial cresce exponencialmente, sendo que a matriz do sistema linear tem $2^{\mathrm{M}} - 1$ linhas e M colunas. Em ambos casos, considerações foram propostas, que diminuem esta complexidade. Uma observação interessante é que a complexidade no cálculo só depende do número de fontes envolvidas e não do comprimento do vetor codificado N_m, $\forall m \in \{1, 2, ..., \mathrm{M}\}$, ou do comprimento do vetor de informação K_m. Em todos os casos a taxa ótima mostra que tão perto da unidade pode ser este valor, de maneira que possa ser possível a recuperação da informação. Tendo em conta isto, a taxa ótima é usada no cálculo do limite virtual de Shannon. Este limite indica o valor até o qual a decodificação conjunta pode atingir uma BER nula. Fica como pendente o desenvolvimento de um algoritmo que atinja o limite virtual de Shannon com o uso da taxa ótima.

No Capítulo 3, a decodificação conjunta em todas as suas variantes mostraram ganhos no seu desempenho sobre a decodificação independente. A complexidade foi tratável mesmo para um número moderado de fontes, sendo estudados casos com 3, 5, 10 e 100 fontes correlacionadas. A complexidade da decodificação conjunta também foi tratada para comprimentos da palavra codificada N igual a 306, 3000 e 30000.

No Capítulo 4, a proposta do algoritmo para o problema CEO mostrou que este em parceria com a decodificação conjunta apresentam um melhor desempenho em comparação ao uso do algoritmo de decodificação independente. Os ganhos obtidos com a decodificação conjunta sobre a decodificação independente são próximos aos obtidos em (Ferrari et al. 2012) e (Ferrari et al. 2014), com a diferença que nesses trabalhos usa-se uma única correlação com a fonte escondida para todas as fontes do modelo, enquanto que nesta tese analisa-se um caso com distintos valores de correlação com a fonte escondida.

5.1 Trabalhos Futuros

Entre os trabalhos futuros temos duas linhas muito diferenciadas.

- A primeira linha de estudo é consequência do visto na Seção 2.3.3, onde apresenta-se um esquema de codificação com taxas $\mathrm{r}_m = \mathrm{K}_m/\mathrm{N}_m$ para um valor ótimo de $\sum_m \mathrm{r}_m$. Neste sentido ficou pendente nesta tese o estudo de um algoritmo de decodificação conjunta que possa ser usado para taxas de decodificação diferenciadas para cada fonte U^m.

- A segunda linha de estudo é consequência do conhecimento do limite virtual de Shannon estudado na Seção 2.4.2. Nesse sentido fica pendente procurar um algoritmo de decodificação ou código que permita chegar muito próximo a este limite. Também fica pendente entender sua relação com outros limites, como o limite da taxa de erro de bit para matrizes $LDGM$ visto em (Garcia-Frias & Zhong 2003).

Apêndice A

A.1 Obtendo a Esperança de U^iU^j para $i \neq j$

Define-se a Z como

$$Z = U^iU^j, \tag{A.1}$$

dado isto se sabe que

$$E[U^iU^j] = E[Z] = 1\ P_z(1) + 0\ P_z(0) \tag{A.2}$$

e

$$P_z(1) = P(u^i = 1, u^j = 1). \tag{A.3}$$

Assim se obtêm que

$$E[U^iU^j] = P(u^i = 1, u^j = 1). \tag{A.4}$$

Reordenando U^i em seus componentes U^0 e E^i do modelo de fonte da Subseção 1.5 como na Figura A.1 e usando o teorema de Bayes se deduz que

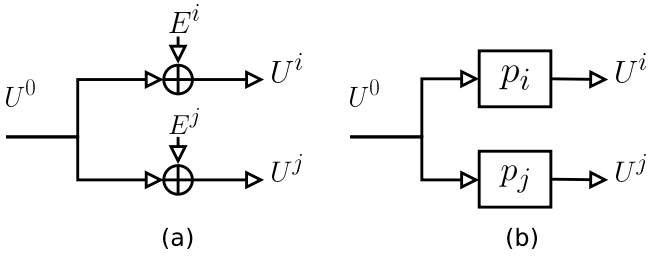

Figura A.1: Modelo de correlação entre U^i e U^j

$$(u^i = 1) \ \equiv \ (u = 1, e^i = 0) \cup (u = 0, e^i = 1), \tag{A.5}$$

$$P(u^i = 1, u^j = 1) = P(u^i = 1/u^j = 1)P(u^j = 1). \tag{A.6}$$

Dado que $(u^0 = 1, e^i = 0)$ e $(u^0 = 0, e^i = 1)$ não tem interseção obtém-se

$$\begin{aligned} P(u^i = 1, u^j = 1) \ &= \ P((u^0 = 1, e^i = 0)/u^j = 1)P(u^j = 1) \\ &+ \ P((u^0 = 0, e^i = 1)/u^j = 1)P(u^j = 1). \end{aligned} \tag{A.7}$$

Usando novamente o teorema de Bayes obtemos

$$\begin{aligned} P(u^i=1,u^j=1) &= P(u^j=1/(u^0=1,e^i=0))P(u^0=1,e^i=0) \\ &+ P(u^j=1/(u^0=0,e^i=1))P(u^0=0,e^i=1). \end{aligned} \tag{A.8}$$

Sabendo que E^i e U^0 são independentes e que U^j não depende de E^i então calcula-se $P(u^i=1,u^j=1)$ como

$$\begin{aligned} P(u^i=1,u^j=1) &= P(e^j=0)P(u^0=1)P(e^i=0) \\ &+ P(e^j=1)P(u^0=0)P(e^i=1), \end{aligned} \tag{A.9}$$

$$E[U^iU^j] = p_0 + p_ip_j - p_0p_i - p_0p_j. \tag{A.10}$$

A.2 Covariância Entre as Fontes U^i e U^j

Conhecendo que

$$cov(U^i,U^j) = E[U^iU^j] - E[U^i]E[U^j] \tag{A.11}$$

e o modelo de geração de fontes correlacionadas da Figura 1.10 e A.1, pode-se deduzir as seguintes relações para todo $i \neq j$:

$$\begin{aligned} P(u^i=1) &= P(e^i=0)P(u^0=1) + P(e^i=1)P(u^0=0) \\ &= p_i + p_0 - 2p_ip_0. \end{aligned} \tag{A.12}$$

Por outro lado, pode-se dizer que $E[U^iU^j]$ é igual a

$$E[U^iU^j] = p_0 + p_ip_j - p_0p_i - p_0p_j \tag{A.13}$$

como pode ser visto detalhadamente no Anexo A.1. Assim obtemos que

$$cov(U^i,U^j) = p_0(1-p_0)(1-2p_i)(1-2p_j), \tag{A.14}$$

$$cov(U^i,U^j) = p_0(1-p_0)[1-2(p_i+p_j-2p_ip_j)]. \tag{A.15}$$

A.3 Correlação Entre as Fontes U^i e U^j

Para poder obter a correlação seguindo o modelo de geração de fontes correlacionadas da Figura 1.10 e A.1, aplicamos que

$$corr(U^i,U^j) = \frac{cov(U^i,U^j)}{\sigma_{u^i}\sigma_{u^j}}, \tag{A.16}$$

$$\sigma_{u^i}^2 = E[{U^i}^2] - E[U^i]^2 = P(u^i=1) - P(u^i=1)^2. \tag{A.17}$$

Usando a equação (A.12)

$$\sigma_{u^i}^2 = p_0(1-p_0) + p_i(1-p_i)(1-2p_0)^2, \tag{A.18}$$

$$corr(U^i,U^j) = \frac{1-2(p_i+p_j-2p_ip_j)}{\sqrt{1+\frac{p_i(1-p_i)(1-2p_0)}{p_0(1-p_0)}}\sqrt{1+\frac{p_j(1-p_j)(1-2p_0)}{p_0(1-p_0)}}}. \tag{A.19}$$

Reordenando definimos que

$$a_i^2 \equiv p_i(1-p_i) \tag{A.20}$$

e

$$b_i \equiv (1-2p_i) \tag{A.21}$$

obtendo

$$corr(U^i, U^j) = \frac{b_i b_j}{\sqrt{1+\frac{a_i^2 b_0^2}{a_0^2}}\sqrt{1+\frac{a_j^2 b_0^2}{a_0^2}}}, \tag{A.22}$$

em geral $corr(U^i, U^j)$ depende só de p_0, p_i e p_j.

A.4 Equivalência de Modelos BSC e XOR

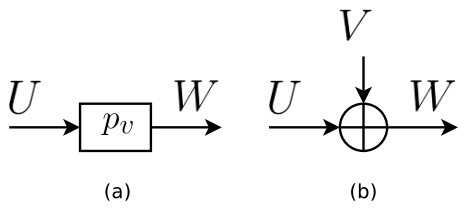

Figura A.2: Equivalência de modelos de soma de ruído

Dada uma fonte U com probabilidade $P(u=1)=p_u$, que percorre um canal BSC com probabilidade de erro p_v, como na Figura A.2 (a), a probabilidade $P(w=1)=p_w$ esta relacionada com p_u e p_v como segue.

$$p_w = P(u=0)P(w=1|u=0) + P(u=1)P(w=1|u=1), \tag{A.23}$$

$$p_w = p_u + p_v - 2p_u p_v. \tag{A.24}$$

No caso da Figura A.2 (b) a probabilidade $P(w=1)=p_w$ acontece quando $u=1$ e $v=0$ ou $u=0$ e $v=1$ sendo as fontes u e v descorrelacionadas, onde

$$p_w = P(u=1)P(v=0) + P(u=0)P(v=1), \tag{A.25}$$

$$p_w = p_u + p_v - 2p_v p_u. \tag{A.26}$$

Como pode-se ver de (A.24) e (A.26), os modelos da Figura A.4 (a) e (b) são equivalentes.

Como notação usarei

$$p_u||p_v \equiv p_u + p_v - 2p_u p_v, \tag{A.27}$$

para obter

$$p_w = p_u||p_v. \tag{A.28}$$

Assim pode-se generalizar que para L fontes descorrelacionadas a_i com probabilidades $P(a_i = 1) = p_{a_i}$, $\forall i \in \{1,2,...,L\}$, que estão sendo ligadas por uma operação XOR, a probabilidade de que o resultado seja de um 1 lógico está dado por

$$b = a_1 \oplus a_2 \oplus ... \oplus a_L, \tag{A.29}$$

$$P(b=1) \equiv p_b = p_{a_1}||p_{a_2}||...||p_{a_L}. \tag{A.30}$$

A.5 Correlação Após um Código $LDPC$

Dadas duas fontes binárias V^i e V^j geradas codificando mediante uma $LDPC$ duas fontes correlacionadas $u^i = u^0 \oplus u^i$ e $u^j = u^0 \oplus u^j$ com $P(e^i = 1) = p_i$, $P(e^j = 1) = p_j$ e $P(u^0 = 1) = 0.5$ como mostra a Figura A.3. Assumindo que a $LDPC$ tem uma quantidade L de uns por coluna

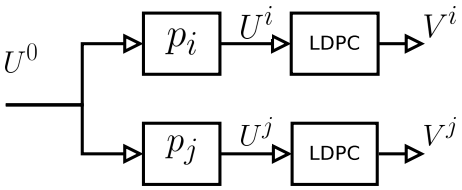

Figura A.3: Gráfico de geração de V^i e V^j

na matriz geradora G e conhecendo a equivalência entre um canal BSC e uma suma XOR visto no Anexo A.4, pode-se reinterpretar a Figura A.3 para obter a Figura A.4, obtendo-se que

$$\begin{array}{rcl} v^i & = & (u^{01} \oplus e^{i1}) \oplus (u^{02} \oplus e^{i2}) \oplus \cdots \oplus (u^{0L} \oplus e^{iL}) \\ & = & (u^{01} \oplus u^{02} \oplus \cdots \oplus u^{0L}) \oplus (e^{i1} \oplus e^{i2} \oplus \cdots \oplus e^{iL}). \end{array} \tag{A.31}$$

Se definimos $u^{0T} \equiv u^{01} \oplus u^{02} \oplus \cdots \oplus u^{0L}$ como uma fonte binária de $P(u^{0T} = 1) = p_{u^{0T}} = 0.5$ e $e^{iT} \equiv e^{i1} \oplus e^{i2} \oplus \cdots \oplus e^{iL}$ como uma fonte binária estatisticamente independente a u^{0T} com probabilidade $P(e^{iT} = 1) = p_{e^{iT}}$ igual a

$$\begin{array}{rcl} p_{e^{iT}} & = & \overbrace{p_i||p_i||...||p_i}^{L}, \\ & \equiv & p_i||^L \end{array} \tag{A.32}$$

Assim obtêm-se

$$v^i = u^{0T} \oplus e^{iT}, \tag{A.33}$$

$$v^j = u^{0T} \oplus e^{jT}, \tag{A.34}$$

$$p_{e^{iT}} = p_i||^L, \tag{A.35}$$

$$p_{e^{jT}} = p_j||^L, \tag{A.36}$$

$$p_{u^{0T}} = 0.5. \tag{A.37}$$

Aplicando o estudado no Anexo A.3, a correlação entre V^i e V^j

$$\begin{array}{rcl} Corr(V^i, V^j) & = & 1 - 2(p_i||^L) \;||\; (p_j||^L) \\ & = & 1 - 2\overbrace{(p_i||p_j) \;||\; (p_i||p_j) \;||...||\; (p_i||p_j)}^{L} \\ & \equiv & 1 - 2(p_i||p_j)||^L \end{array} \tag{A.38}$$

Na equação (A.38) o termo $(p_i||p_j)||^L$ tende a 0.5 com o aumento de valor de L. Conhecendo isto se deduz que a correlação tenderá a zero com o aumento de valor de L.

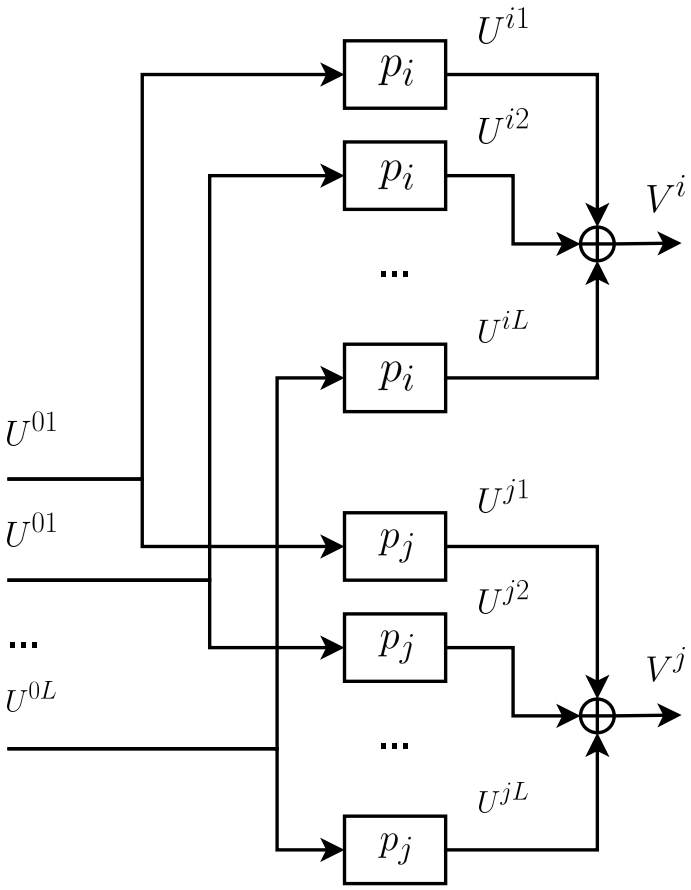

Figura A.4: Gráfico de geração de um bit codificado V_1

A.6 Obtendo a Covariância Entre a Entrada e a Saída de um Canal BSC

Dadas duas fontes binárias U^i e U^j, onde U^j é gerado enviando os dados de U^i por um canal BSC com probabilidade de erro p_{ij}, como na Figura A.5, a correlação é expressada como

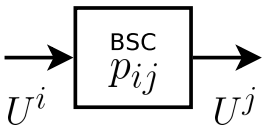

Figura A.5: Gráfico de U^j gerado enviando os dados de U^i por um canal BSC com probabilidade de erro p_{ij}.

$$cov(U^i, U^j) = E[U^i U^j] - E[U^i]E[U^j]. \tag{A.39}$$

Conhecendo que $E[U^i] = P(u^i = 1)$,$E[U^j] = P(u^j = 1)$ e (A.4) e substituindo em (A.39) obtemos

$$cov(U^i, U^j) = P(u^i = 1, u^j = 1) - P(u^i = 1)P(u^j = 1), \tag{A.40}$$

$$cov(U^i, U^j) = P(u^j = 1|u^i = 1)P(u^i = 1) - P(u^i = 1)P(u^j = 1), \tag{A.41}$$

$$cov(U^i, U^j) = P(u^i = 1)[(1 - p_{ij}) - P(u^j = 1)], \tag{A.42}$$

$$cov(U^i, U^j) = P(u^i = 1)(1 - P(u^j = 1))\{1 - \frac{p_{ij}}{1 - P(u^j = 1)}\}. \tag{A.43}$$

A.7 Obtendo a Correlação Entre a Entrada e a Saída de um Canal BSC

Conhecendo a equação (A.43) e a equação (A.17)

$$corr(U^i, U^j) = \frac{1 - \frac{p_{ij}}{1-P(u^j=1)}}{\sqrt{\frac{1-P(u^i=1)}{P(u^i=1)}}\sqrt{\frac{P(u^j=1)}{(1-P(u^j=1))}}}. \tag{A.44}$$

A.8 Obtendo a Probabilidade $P(U^1U^2U^3 \ldots U^{\mathrm{M}}|U^0)$

Dado um conjunto de fontes binárias $U^i \ \forall i \in \{1, 2, \ldots, M\}$, onde U^i é gerado enviando os dados de uma fonte U^0 de $P(u^0 = 1) = p_0$ por um canal BSC com probabilidade de erro p_i, como na Figura 1.10. O conhecimento de U^0 faz que os valores de $U^1, U^2, U^3, \ldots, U^{\mathrm{M}}$ dependa só dos valores de $E^1, E^2, E^3, \ldots, E^{\mathrm{M}}$ respectivamente, sendo estos últimos estatisticamente independentes, gerando as seguintes relações

$$P(U^1U^2U^3 \ldots U^{\mathrm{M}}|U^0) = P(U^1|U^0)P(U^2|U^0)P(U^3|U^0) \ldots P(U^{\mathrm{M}}|U^0). \tag{A.45}$$

Definindo

$$P(U^i|U^0) = b_i = \begin{cases} p_i & \text{if } u^i \neq u^0 \\ 1 - p_i & \text{if } u^i = u^0 \end{cases} \tag{A.46}$$

obtemos

$$P(U^1U^2U^3 \ldots U^{\mathrm{M}}|U^0) = b_1b_2b_3 \ldots b_m. \tag{A.47}$$

A.9 Obtendo a Probabilidade $P(U^0|U^1U^2U^3 \ldots U^{\mathrm{M}})$

Dado um conjunto de fontes binárias $U^i \ \forall i \in \{1, 2, \ldots, \mathrm{M}\}$, onde U^i é gerado enviando os dados de uma fonte U^0 de $P(u^0 = 1) = p_0$ por um canal BSC com probabilidade de erro p_i, como na Figura 1.10. Para obter a probabilidade de que U^0 tenha um valor a, $a = \{0, 1\}$, conhecendo o valor atual das fontes $U^1, U^2, U^3, \ldots, U^{\mathrm{M}}$ calcula-se

$$\begin{aligned} P(U^0|U^1U^2 \ldots U^{\mathrm{M}}) &= \frac{P(U^1U^2 \ldots U^{\mathrm{M}}|U^0)P(U^0)}{P(U^1U^2U^3 \ldots U^{\mathrm{M}})} \\ &= \frac{b_1b_2b_3 \ldots b_m P(U^0)}{P(U^1U^2 \ldots U^{\mathrm{M}})} \end{aligned}. \tag{A.48}$$

A.10 Obtendo a Probabilidade $P(U^1U^2U^3\dots U^{\mathrm{M}})$

Dado um conjunto de fontes binárias $U^i\ \forall i \in \{1,2,\dots,\mathrm{M}\}$, onde U^i é gerado enviando os dados de uma fonte U^0 de $P(u^0=1)=p_0$ por um canal BSC com probabilidade de erro p_i, como pode-se ver na Figura 1.10. A probabilidade conjunta é dada por

$$\begin{aligned} P(U^1U^2\dots U^{\mathrm{M}}) &= P(U^1U^2\dots U^{\mathrm{M}}|u^0=0)P(u^0=0) \\ &+ P(U^1U^2\dots U^{\mathrm{M}}|u^0=1)P(u^0=1). \end{aligned} \tag{A.49}$$

O conhecimento de U^0 faz as probabilidades de $U^1,U^1,\dots,U^{\mathrm{M}}$ dependentes só de o ruído do canal que percorrem. Assim a equação (A.49) seria rescrita como

$$\begin{aligned} P(U^1U^2\dots U^{\mathrm{M}}) &= P(U^1|u^0=0)P(U^2|u^0=0)\dots P(U^{\mathrm{M}}|u^0=0)P(u^0=0) \\ &+ P(U^1|u^0=1)P(U^2|u^0=1)\dots P(U^{\mathrm{M}}|u^0=1)P(u^0=1). \end{aligned} \tag{A.50}$$

Sabendo que $P(u^0=1)=p_0$ e p_i é a probabilidade de erro do canal BSC que percorre a informação da fonte U^i, usamos

$$P(U^i|u^0=0)=a_i=\begin{cases} p_i & \text{if } u^i=1 \\ 1-p_i & \text{if } u^i=0 \end{cases} \tag{A.51}$$

para modificar a equação (A.50), obtendo

$$P(U^1U^2\dots U^{\mathrm{M}})=a_1a_2\dots a_m(1-p_0)+(1-a_1)(1-a_2)\dots(1-a_m)p_0. \tag{A.52}$$

A.11 Obtendo a Probabilidade $P(U(S)|U(S^c))$

Dado um conjunto de fontes binárias U^i, $\forall i \in \{1,2,\dots,\mathrm{M}\}$, onde U^i é gerado enviando os dados de uma fonte U^0 de $P(u^0=1)=p_0$ por um canal BSC com probabilidade de erro p_i, como pode-se ver na Figura 1.10. Se define $S \subseteq \{1,2,...,\mathrm{M}\}$, S^c como seu complemento e $S^t=\{1,2,...,\mathrm{M}\}$. Sendo

$$S^t = S \bigcup S^c. \tag{A.53}$$

Se define

$$U(S)=\bigcup_{i\in S} U^i \tag{A.54}$$

Aplicando o teorema de Bayes e a equação (A.53) sabe-se que

$$P(S^t)=P(S\bigcup S^c)=P(S|S^c)P(S^c), \tag{A.55}$$

onde obtemos

$$P(S|S^c)=\frac{P(S^t)}{P(S^c)}=\frac{P(U^1U^1\dots U^{\mathrm{M}})}{P(S^c)}. \tag{A.56}$$

Isto pode ser facilmente calculado aplicando o Apêndice A.10

$$a(S)=\prod_{i\in S} a_i, \tag{A.57}$$

$$a'(S)=\prod_{i\in S}(1-a_i), \tag{A.58}$$

$$P(S|S^c)=\frac{a(S^t)(1-p_0)+a'(S^t)(p_0)}{a(S^c)(1-p_0)+a'(S^c)(p_0)}. \tag{A.59}$$

A.12 Obtendo a Informação $H(U^1U^2...U^{\mathrm{M}}|U^0)$

Dado um conjunto de fontes binárias U^i, $\forall i \in \{1, 2, \ldots, \mathrm{M}\}$, como na Figura 1.10. U^i é gerado enviando os dados de uma fonte U^0 de $P(u^0 = 1) = p_0$ por um canal BSC com probabilidade de erro p_i. $H(U^1U^2...U^{\mathrm{M}}|U^0)$ pode ser expressado com

$$\begin{aligned} H(U^1U^2...U^{\mathrm{M}}|U^0) = & \quad H(U^1|U^0) \\ & + \quad H(U^2|U^1U^0) \\ & + \quad H(U^3|U^2U^1U^0) \\ & \quad \cdots \\ & + \quad H(U^{\mathrm{M}}|U^{\mathrm{M}-1}\ldots U^1U^0). \end{aligned} \tag{A.60}$$

Sabendo que o conhecimento de U^0 faz que U^i dependa só da fonte E^i

$$u^i = e^i \; XOR \; u^0, \tag{A.61}$$

$$\begin{aligned} H(U^1U^2...U^{\mathrm{M}}|U^0) = & \quad H(U^1|U^0) \\ & + \quad H(U^2|E^1U^0) \\ & + \quad H(U^3|E^2E^1U^0) \\ & \quad \cdots \\ & + \quad H(U^{\mathrm{M}}|E^{\mathrm{M}-1}\ldots E^1U^0) \end{aligned} \quad , \tag{A.62}$$

$$\begin{aligned} H(U^1U^2...U^{\mathrm{M}}|U^0) = & \quad H(U^1|U^0) \\ & + \quad H(U^2|U^0) \\ & + \quad H(U^3|U^0), \\ & \quad \cdots \\ & + \quad H(U^{\mathrm{M}}|U^0) \end{aligned} \tag{A.63}$$

$$H(U^1U^2...U^{\mathrm{M}}|U^0) = \sum_{i=1}^{\mathrm{M}} H(U^i|U^0), \tag{A.64}$$

$$H(U^1U^2...U^{\mathrm{M}}|U^0) = \sum_{i=1}^{\mathrm{M}} h(p_i). \tag{A.65}$$

Onde $h(p_i)$ é a entropia binária da probabilidade p_i.

A.13 Obtendo a Informação $H(U^0U^1...U^{\mathrm{M}})$

Dado um conjunto de fontes binárias U^i, $\forall i \in \{1, 2, \ldots, \mathrm{M}\}$, onde U^i é gerado enviando os dados de uma fonte U^0 de $P(u^0 = 1) = p_0$ por um canal BSC com probabilidade de erro p_i, como pode-se ver na Figura 1.10. $H(U^0U^1...U^{\mathrm{M}})$ se expressa

$$H(U^0U^1...U^{\mathrm{M}}) = H(U^1U^2...U^{\mathrm{M}}|U^0) + H(U^0). \tag{A.66}$$

Usando o apêndice A.12

$$H(U^0U^1...U^{\mathrm{M}}) = \sum_{i=1}^{\mathrm{M}} h(p_i) + H(U^0). \tag{A.67}$$

A.14 Obtendo a Informação $H(U^0|U^1U^2...U^{\mathrm{M}})$

Nesta seção assume-se que tem-se um conjunto de fontes binárias U^i, $\forall i \in \{1, 2, \ldots, \mathrm{M}\}$, onde U^i é gerado enviando os dados de uma fonte U^0 de $P(u^0 = 1) = p_0$ por um canal BSC com probabilidade de erro p_i, como pode-se ver Figura 1.10.

Antes de iniciar é necessário indicar que $H(U^0|U^1U^2...U^{\mathrm{M}})$ expressa a informação de U^0 dado que se conhece $U^1U^2...U^{\mathrm{M}}$. Assim $H(U^0|U^1U^2...U^{\mathrm{M}})$ é o que não se conhece de U^0 quando sabemos o estado atual das fontes U^i.

Explicado tudo isto $H(U^0|U^1U^2...U^{\mathrm{M}})$ expressa-se como

$$H(U^0|U^1U^2...U^{\mathrm{M}}) = H(U^0U^1...U^{\mathrm{M}}) - H(U^1U^2...U^{\mathrm{M}}). \tag{A.68}$$

Usando a equação (A.66) na equação anterior

$$H(U^0|U^1U^2...U^{\mathrm{M}}) = H(U^1U^2...U^{\mathrm{M}}|U^0) + H(U^0) - H(U^1U^2...U^{\mathrm{M}}). \tag{A.69}$$

Incluindo o resultado da equação (A.64), obtemos

$$H(U^0|U^1U^2...U^{\mathrm{M}}) = \sum_{i=1}^{\mathrm{M}} H(U^i|U^0) + H(U^0) - H(U^1U^2...U^{\mathrm{M}}), \tag{A.70}$$

$$H(U^0|U^1U^2...U^{\mathrm{M}}) = \sum_{i=1}^{\mathrm{M}} h(p_i) + h(p_0) - H(U^1U^2...U^{\mathrm{M}}). \tag{A.71}$$

A.15 Demonstração de Taxa Ótima para um Caso Específico

Dada uma função objetivo

$$r_{max} = \max\{r\}. \tag{A.72}$$

Sendo que r tem as seguintes restrições

$$r \leq \frac{|S|C}{H(U(S)|U(S^c))}. \tag{A.73}$$

Onde $|S|$ é o numero de fontes envolvidas no subconjunto S. Para resolver estas equações é necessário provar que

$$\frac{M}{H(U^1U^2...U^{\mathrm{M}})} \leq \frac{|S|}{H(U(S)|U(S^c))} \tag{A.74}$$

conhecendo que $|S| + |S^c| = M$, $H(U(S)|U(S^c)) = H(U^1U^2...U^{\mathrm{M}}) - H(U(S^c))$, pode-se dizer que (A.74) é equivalente a provar que

$$\frac{H(U^1U^2...U^{\mathrm{M}})}{M} \leq \frac{H(U(S^c))}{|S^c|} \tag{A.75}$$

ou

$$\frac{H(U^1U^2...U^{\mathrm{M}})}{M} \leq \frac{H(U(S))}{|S|}. \tag{A.76}$$

Por simplicidade de notação a equação (A.76) é transformada em

$$\frac{H(M)}{M} \leq \frac{H(|S|)}{|S|} \tag{A.77}$$

Assumindo um caso mais restritivo que na equação (A.77), onde $m = |S|$

$$\frac{H(m)}{m} \leq \frac{H(m-1)}{m-1}. \tag{A.78}$$

Usando a regra da cadeia em $H(m) = H(1|m-1) + H(m-1)$ a equação (A.78) é equivalente a

$$H(1|m-1) \leq \frac{H(m-1)}{m-1}, \tag{A.79}$$

além disso é possível rescrever $H(m-1)$ da seguinte maneira

$$\begin{array}{rc} H(m-1) & = H(1|m-2) \\ & +... \\ & +H(1|1) \\ & +H(1) \end{array} \tag{A.80}$$

pode-se dizer que

$$(m-1)H(1|m-2) \leq H(m-1) \tag{A.81}$$

ou

$$H(1|m-2) \leq \frac{H(m-1)}{(m-1)} \tag{A.82}$$

conhecendo da última equação que $\frac{H(m-1)}{m-1}$ sempre sera maior que $H(1|m-2)$ e misturando esta ideia com a equação (A.79), se obtêm

$$H(1|m-1) \leq H(1|m-2). \tag{A.83}$$

Dado que isto é verdade, pode-se deduzir que a equação (A.74) é verdadeira. Misturando as equações (A.72),(A.73) e (A.74) é obtido.

$$r_{max} = \frac{MC}{H(U^1U^2...U^{\mathrm{M}})}. \tag{A.84}$$

A inequação (A.77) pode ser deduzida utilizando-se o Teorema 17.6.1 de (Cover & Thomas 2006) que foi originalmente proposto em (Han 1978). É fácil de ver na definição 17.51 de (Cover & Thomas 2006) que todos os termos do somatório são idênticos e assim a definição se transforma no lado esquerdo da inequação (A.77). Esta observação foi sugerida por um revisor anônimo que também chamou a atenção para o fato de que o Teorema 17.6.1 pode ser deduzido considerando propriedades de sub-modularidade da função de entropia (Tian 2011). É importante observar que as inequações continuam válidas para a entropia diferencial.

A.16 Aproximação da Entropia Binária

Dada a função

$$f(x) = h(1/2 - x) \tag{A.85}$$

onde $h(a)$ é a entropia binária $h(a) = -alog_2(a) - (1-a)log_2(1-a)$. Aplica-se a serie de Taylor ao redor de $x = 0$ para aproximar a função $f(x)$, obtendo

$$f(x) = f(0) + f'(0)x + f''(0)\frac{x^2}{2} + ... \tag{A.86}$$

Sendo

$$f'(x) = \frac{df(x)}{dx} = log_2(\frac{1/2 - x}{1/2 + x}) \tag{A.87}$$

e

$$f''(x) = \frac{d^2f(x)}{dx^2} = \frac{-1}{(1/2 + x)(1/2 - x)ln(2)}. \tag{A.88}$$

A função $f(x)$ quedaria expressada como

$$f(x) = 1 - \frac{2}{ln(2)}x^2 + ... \tag{A.89}$$

Ficando a entropia binária aproximadamente igual a

$$h(\frac{1}{2} - x) \approx 1 - \frac{2}{ln(2)}x^2 \tag{A.90}$$

quando os valores de x estão próximos ao redor de $x = 0$.

A.17 Aproximação da Função $Q(x)$

Conhecendo que a função $Q(x)$ está definida como

$$Q(x) = \frac{1}{2} - \frac{1}{\sqrt{2\pi}}\int_0^x e^{-a^2/2}da. \tag{A.91}$$

Usa-se a serie de Taylor de

$$e^{-a^2/2} = \sum_{n=0}^{\infty}\frac{(-1)^n a^{2n}}{2^n n!} \tag{A.92}$$

para valores próximos ao redor de $a = 0$, para obter a serie de Taylor de $Q(x)$ como

$$Q(x) = \frac{1}{2} - \frac{1}{\sqrt{2\pi}}\sum_{n=0}^{\infty}\{\frac{(-1)^n}{2^n n!}\int_0^x a^{2n}da\}. \tag{A.93}$$

$$Q(x) = \frac{1}{2} - \frac{x}{\sqrt{2\pi}} + \frac{x^3}{6\sqrt{2\pi}} + ... \tag{A.94}$$

Ficando $Q(x)$ aproximadamente igual a

$$Q(x) \approx \frac{1}{2} - \frac{x}{\sqrt{2\pi}} \tag{A.95}$$

quando os valores de x estão próximos ao redor de $x = 0$.

A.18 Aproximação da Capacidade de Canal de um Canal BSC

A capacidade de canal de um canal BSC esta dado por

$$C_{BSC} = 1 - h(Q(\sqrt{2rE_b/N_0})), \tag{A.96}$$

sendo r a taxa de codificação. Se r tende a um valor próximo a zero, então usando as equações (A.90) e (A.95) se obtém a seguinte aproximação

$$C_{BSC} = \frac{2rE_b/N_0}{ln(2)\pi}. \tag{A.97}$$

A.19 Limite de Shannon de um Canal BSC para uma Taxa de Codificação r Tendendo a Zero

O limite de Shannon é achado igualando a capacidade de canal C_{BSC} com a taxa de informação r enviada pelo canal. Para o caso especial em que a taxa de informação r e muito próximo a zero, pode-se usar a equação (A.97) obtendo-se

$$r = \frac{2rE_b/N_0}{ln(2)\pi}, \tag{A.98}$$

de modo que

$$E_b/N_0 = \frac{ln(2)\pi}{2}. \tag{A.99}$$

A.20 Limite de Shannon de um Canal $BI\text{-}AWGN$ para uma Taxa de Codificação r Tendendo a Zero

A equação (1.28) para o limite de Shannon de um canal $BI\text{-}AWGN$ é reordenada usando $a = y\sqrt{2N_0}$ e $b = \sqrt{2rE_b/N_0}$ obtendo-se

$$\mathrm{C}_{BI-AWGN} = \frac{1}{2\sqrt{2\pi}} \int_{-\infty}^{+\infty} (e^{\frac{-(a-b)^2}{2}} log_2(\frac{2}{1+e^{-2ab}}) + e^{\frac{-(a+b)^2}{2}} log_2(\frac{2}{1+e^{+2ab}}))da, \tag{A.100}$$

com

$$r = \frac{b^2}{2E_b/N_0}. \tag{A.101}$$

Se igualarmos a taxa do código r e a capacidade de canal C obtém-se

$$\frac{b^2}{2E_b/N_0} = \frac{1}{2\sqrt{2\pi}} \int_{-\infty}^{+\infty} (f(a,b) + f(a,-b))da, \tag{A.102}$$

$$f(a,b) = e^{\frac{-(a-b)^2}{2}} log_2(\frac{2}{1+e^{-2ab}}). \tag{A.103}$$

Dado que ambos membros da equação (A.102) tendem a zero quando r tende a zero (ou seja b tende a zero), para poder obter o valor limite de E_b/N_0 pode-se derivar ambos extremos da equação (A.102) em relação a b, obtendo-se.

$$\frac{b}{E_b/N_0} = \frac{1}{2\sqrt{2\pi}} \int_{-\infty}^{+\infty} (f'(a,b) - f'(a,-b))da. \tag{A.104}$$

Avaliando $f'(a,b) - f'(a,-b)$ em $b = 0$ obtêm-se o valor de zero. Dado isto se deriva novamente a equação (A.104) e se avalia em $b = 0$ para obter

$$\frac{1}{E_b/N_0} = \frac{1}{2\sqrt{2\pi}} \int_{-\infty}^{+\infty} (2f''(a,0))da. \tag{A.105}$$

Sabendo que

$$\frac{\partial(e^{\frac{-(a-b)^2}{2}})}{\partial b} = (a-b)e^{\frac{-(a-b)^2}{2}}, \tag{A.106}$$

$$\frac{\partial(log_2(\frac{2}{1+e^{-2ab}}))}{\partial b} = \frac{2a}{ln(2)(1+e^{+2ab})}, \tag{A.107}$$

$$f'(a,b) = (a-b)e^{\frac{-(a-b)^2}{2}} log_2(\frac{2}{1+e^{-2ab}}) + e^{\frac{-(a-b)^2}{2}} \frac{2a}{ln(2)(1+e^{+2ab})}, \tag{A.108}$$

$$f'(a,b) = (a-b)f(a,b) + \frac{2a}{ln(2)} \frac{e^{\frac{-(a-b)^2}{2}}}{(1+e^{+2ab})}. \tag{A.109}$$

Como dados adicionais se calcula que

$$f(a,0) = 0, \quad f'(a,0) = \frac{ae^{\frac{-a^2}{2}}}{ln(2)}. \tag{A.110}$$

A segunda derivada de $f(a,b)$ é

$$f''(a,b) = -f(a,b) + (a-b)f'(a,b) + \frac{2a}{ln(2)} \frac{(a-b)e^{\frac{-(a-b)^2}{2}}}{(1+e^{+2ab})} - \frac{4a^2}{ln(2)} \frac{e^{\frac{-(a-b)^2}{2}+2ab}}{(1+e^{+2ab})^2}, \tag{A.111}$$

$$f''(a,0) = +\frac{a^2 e^{\frac{-a^2}{2}}}{ln(2)}. \tag{A.112}$$

Reformulando a equação (A.105)

$$\frac{1}{E_b/N_0} = \frac{1}{ln(2)\sqrt{2\pi}} \int_{-\infty}^{+\infty} a^2 e^{\frac{-a^2}{2}} da, \tag{A.113}$$

$$\frac{1}{E_b/N_0} = \frac{1}{ln(2)\sqrt{2\pi}} \int_{-\infty}^{+\infty} e^{\frac{-a^2}{2}} da - \frac{a}{ln(2)\sqrt{2\pi}e^{\frac{a^2}{2}}}|_{-\infty}^{+\infty}, \tag{A.114}$$

$$E_b/N_0 = ln(2). \tag{A.115}$$

Bibliografia

Abrardo, A., Ferrari, G. & Martalo, M. (2011). On non-cooperative block-faded orthogonal multiple access schemes with correlated sources, *Communications, IEEE Transactions on* **59**(7): 1916–1926.

Abrardo, A., Ferrari, G., Martalò, M., Franceschini, M. & Raheli, R. (2012). Orthogonal multiple access with correlated sources: Feasible region and pragmatic schemes, *CoRR* **abs/1201.6548**.

Barros, J. & Servetto, S. (2006). Network information flow with correlated sources, *Information Theory, IEEE Transactions on* **52**(1): 155–170.

Barros, J. & Tuchler, M. (2006). Scalable decoding on factor trees: a practical solution for wireless sensor networks, *Communications, IEEE Transactions on* **54**(2): 284–294.

Cover, T. M. & Thomas, J. A. (2006). *Elements of information theory (2. ed.)*, Wiley.

Del Ser, J., Garcia-Frias, J. & Crespo, P. M. (2009). Iterative concatenated zigzag decoding and blind data fusion of correlated sensors, *Ultra Modern Telecommunications Workshops, 2009. ICUMT '09. International Conference on*, pp. 1–6.

Ferrari, G., Martalo, M., Abrardo, A. & Raheli, R. (2012). Orthogonal multiple access and information fusion: How many observations are needed?, pp. 311–320.

Ferrari, G., Martalò, M. & Abrardo, A. (2014). Information fusion in wireless sensor networks with source correlation, *Information Fusion* **15**(0): 80 – 89. Special Issue: Resource Constrained Networks.
*http://www.sciencedirect.com/science/article/pii/S156625351200084X

Gallager, R. (1962). Low-density parity-check codes, *Information Theory, IRE Transactions on* **8**(1): 21–28.

Garcia-Frias, J., Zhao, Y. & Zhong, W. (2007). Turbo-like codes for transmission of correlated sources over noisy channels, *Signal Processing Magazine, IEEE* **24**(5): 58–66.

Garcia-Frias, J. & Zhong, W. (2003). Approaching shannon performance by iterative decoding of linear codes with low-density generator matrix, *Communications Letters, IEEE* **7**(6): 266–268.

GPLK (2000). The gplk (gnu linear programming kit) package. *https://www.gnu.org/software/glpk/

Haghighat, J., Behroozi, H. & Plant, D. (2008). Iterative joint decoding for sensor networks with binary ceo model, pp. 41–45.

Han, T. S. (1978). Nonnegative entropy measures of multivariate symmetric correlations, *Information and Control* pp. 133–156.

Jiang, M., Zhao, C., Shi, Z. & Chen, Y. (2005). An improvement on the modified weighted bit flipping decoding algorithm for ldpc codes, *Communications Letters, IEEE* **9**(9): 814–816.

K. Kobayashi, T. Y. & Katayama, M. (2009). Decoding of separately encoded multiple correlated sources transmitted over noisy channels, *IEICE Trans. Fundamentals* **E92-A**(10): 2402–2410.

Kou, Y., Lin, S. & Fossorier, M. (2001). Low-density parity-check codes based on finite geometries: a rediscovery and new results, *Information Theory, IEEE Transactions on* **47**(7): 2711–2736.

Kschischang, F., Frey, B. & Loeliger, H.-A. (2001). Factor graphs and the sum-product algorithm, *Information Theory, IEEE Transactions on* **47**(2): 498–519.

MacKay, D. J. C. (1999). Good error-correcting codes based on very sparse matrices, *Information Theory, IEEE Transactions on* **45**(2): 399–431.

Pradhan, S. & Ramchandran, K. (2003). Distributed source coding using syndromes (discus): design and construction, *Information Theory, IEEE Transactions on* **49**(3): 626–643.

Pujaico Rivera, F. (2011). *Algoritmos de decodificação abrupta para códigos ldgm*, Master's thesis, Universidade Estadual de Campinas, Campinas - Brasil.

Sartipi, M. & Fekri, F. (2008). Distributed source coding using short to moderate length rate-compatible ldpc codes: the entire slepian-wolf rate region, *Communications, IEEE Transactions on* **56**(3): 400–411.

Schonberg, D., Ramchandran, K. & Pradhan, S. (2004). Distributed code constructions for the entire slepian-wolf rate region for arbitrarily correlated sources, pp. 292–301.

Shamai, S. & Verdu, S. (1995). Capacity of channels with uncoded side information, *European Trans. Telecommun.* **6**(5): 587–600.

Slepian, D. & Wolf, J. (1973a). A coding theorem for multiple access channels with correlated sources, *Bell System Technical Journal, The* **52**(7): 1037–1076.

Slepian, D. & Wolf, J. (1973b). Noiseless coding of correlated information sources, *Information Theory, IEEE Transactions on* **19**(4): 471–480.

Tian, C. (2011). Inequalities for entropies of sets of subsets of random variables, *IEEE International Symposium on Information Theory* pp. 1950 – 1954.

Wu, X., Zhao, C. & You, X. (2007). Parallel weighted bit-flipping decoding, *Communications Letters, IEEE* **11**(8): 671–673.

Yedla, A., Pfister, H. & Narayanan, K. (2013). Code design for the noisy slepian-wolf problem, *Communications, IEEE Transactions on* **61**(6): 2535–2545.