SABRINA FERREIRA MARCIANO FARIA

CÓDIGOS PARA CANAIS DE LEITURA DE PARES DE SÍMBOLOS E DE B-SÍMBOLOS

Dissertação apresentada à Universidade Federal de Viçosa, como parte das exigências do Programa de Pós-Graduação em Matemática, para obtenção do título de *Magister Scientiae*.

VIÇOSA
MINAS GERAIS - BRASIL
2019

Ficha catalográfica preparada pela Biblioteca Central da Universidade Federal de Viçosa - Câmpus Viçosa

T
F224c
2019

Faria, Sabrina Ferreira Marciano, 1993-
Códigos para canais de leitura de pares de símbolos e de b-símbolos / Sabrina Ferreira Marciano Faria. – Viçosa, MG, 2019.
viii, 112 f. : il. (algumas color.) ; 29 cm.

Orientador: Marines Guerreiro.
Dissertação (mestrado) - Universidade Federal de Viçosa.
Referências bibliográficas: f. 110-112.

1. Códigos corretores de erros (Teoria da informação). 2. Teoria da codificação. 3. Lógica simbólica e matemática. I. Universidade Federal de Viçosa. Departamento de Matemática. Programa de Pós-Graduação em Matemática. II. Título.

CDD 22. ed. 519.7

SABRINA FERREIRA MARCIANO FARIA

CÓDIGOS PARA CANAIS DE LEITURA DE PARES DE SÍMBOLOS E DE B-SÍMBOLOS

Dissertação apresentada à Universidade Federal de Viçosa, como parte das exigências do Programa de Pós-Graduação em Matemática, para obtenção do título de *Magister Scientiae*.

APROVADA: 17 de julho de 2019.

Marcelo Firer

Sônia Maria Fernandes

Marinês Guerreiro
(Orientadora)

Dedico este trabalho à minha família e amigos, em especial aos meus pais, Maria Domingas e Joaquim.

"Você nunca sabe que resultados virão da sua ação. Mas se você não fizer nada, não existirão resultados."(Mahatma Gandhi)

Agradecimentos

Primeiramente, agradeço a Deus, por todas as suas bençãos em minha vida, sem ele nada teria alcançado e nada seria.

Aos meus pais, pelo exemplo de vida, pelo apoio incondicional e por estarem presentes em cada passo, em cada tropeço e, principalmente, em cada vitória da minha caminhada até o presente momento. A minha irmã, pelo incentivo.

Ao meu namorado, por estar ao meu lado me dando todo o suporte necessário durante a realização deste trabalho.

A minha orientadora Marinês, por todo o conhecimento transmitido e por acreditar em mim até mesmo quando eu não acreditava.

Aos professores e funcionários do DMA, pelo excelente trabalho prestado e por fazer parte da minha formação.

Aos meus amigos, pelos momentos de descontração e pelos momentos de estudos que, com certeza, auxiliaram em meu aprendizado.

À CAPES, pelo auxilio financeiro imprescindível para a realização deste trabalho.

E finalmente, agradeço a todos que tenham contribuído direta ou indiretamente com a realização deste trabalho.

Sumário

Resumo

FARIA, Sabrina Ferreira Marciano, M.Sc., Universidade Federal de Viçosa, julho de 2019. **Códigos para Canais de Leitura de Pares de Símbolos e de b-Símbolos**. Orientadora: Marinês Guerreiro.

Neste trabalho estudamos duas novas classes de códigos, os *códigos para canais de leitura de pares de símbolos* proposta por Cassuto e Blaum, em 2011, e *os códigos para canais de leitura de b-símbolos* proposto por Yaakobi, Bruck e Siegel, em 2012, nos quais a leitura é feita em blocos de símbolos consecutivos e não símbolo a símbolo como na Teoria Clássica dos Códigos Corretores de Erros. O principal objetivo deste trabalho é fazer um paralelo entre estas duas novas teorias e a teoria clássica. Apresentamos as principais definições e resultados e ressaltamos especialmente a relação entre a distância mínima de Hamming e as distâncias mínimas de pares e de b-símbolos. Além disso, apresentamos alguns algoritmos de decodificação para os códigos para canais de leituras de pares de símbolos.

Abstract

FARIA, Sabrina Ferreira Marciano, M.Sc., Universidade Federal de Viçosa, July, 2019 **Codes for Symbol Pair and b-Symbols Read Channels**. Adviser: Marinês Guerreiro.

In this paper we study two new groups of codes, the *codes for symbol-pair read channels* presented by Cassuto and Blaum in 2011 and *the codes for b-symbols read channels* presented by Yaakobi, Bruck and Siegel in 2012, in which the reading is made in pairs or in blocks of b consecutive symbols and not by single symbols as in the Classical Theory of Error-Correcting Codes. The main goal of this work is to make a parallel between this two new theories and the Classical Theory. We present the main definitions and results and we highlight the relationship between the minimum Hamming distance and the minimum distances of symbol-pair and b-symbol codes. We also discuss some decoding algorithms to the codes for symbol-pair read channels .

Introdução

No nosso cotidiano estamos sempre transmitindo informações, seja uma mensagem de texto para um amigo, um *e-mail* de trabalho ou um recadinho na geladeira para um familiar; recebendo informações, seja assistindo televisão, ouvindo uma música ou recebendo uma ligação; e guardando informações, como por exemplo uma foto salva na nuvem, ou um arquivo de texto salvo no computador. No entanto, durante essas transmissões de informação podem ocorrer interferências, que chamamos de *ruídos*, como interferências electromagnéticas ou um erro humano (por exemplo, erro de digitação) e elas fazem com que a mensagem recebida não seja a mesma que a enviada.

O principal objetivo da Teoria de Códigos Corretores de Erros é detectar e corrigir o maior número de erros possíveis, mas também tem como finalidade transmitir a informação da melhor forma possível para que se possa fazer tais procedimentos de forma eficiente.

A Teoria de Códigos Corretores de Erros foi iniciada pelo matemático C. E. Shannon, no Laborátório de Bell em 1948. Ela é desenvolvida principalmente pela Matemática, Engenharia, Computação e Estatística. É uma teoria em constante desenvolvimento devido à grande demanda da tecnologia atual para que a transmissão de informações seja cada vez mais rápida e segura.

Tradicionalmente, na Teoria da Informação, se analisam mensagens com ruídos em unidades de informação individual, chamadas de *símbolos*, isto é, o processo de escrita e leitura é executado símbolo à símbolo. Contudo, em algumas tecnologias de armazenamento atuais, bem como em algumas propostas para o futuro, os símbolos podem não ser necessariamente escritos e lidos individualmente e sim em grupos sobrepostos. Isto levou Cassuto e Blaum [3], em 2011, a estabelecerem uma nova estrutura de codificação, os *códigos para canais de leitura de pares de símbolos* e, em 2012, Yaakobi, Bruck e Siegel [29, 28] generalizaram esta ideia definindo os *códigos de leitura de b-símbolos*.

O objetivo deste trabalho é estudar os *códigos para canais de leitura de pares de símbolos e de b-símbolos*, fazendo um paralelo com a Teoria Clássica dos Códigos Corretores de Erros, a partir de uma revisão de literatura sobre o tema.

No Capítulo 2, apresentamos a construção desta nova classe de códigos, os códigos de pares de símbolos, como em [3]. Definimos o que é um vetor de pares, a distância de pares (ou métrica de pares) e, num primeiro resultado, comparamos a distância de pares com a distância de Hamming. Para quaisquer dois vetores (ou palavras) $x, y \in \mathcal{A}^n$, com $0 < d_H(x, y) < n$, tem-se

$$d_H(x, y) + 1 \leq d_P(x, y) \leq 2 \cdot d_H(x, y),$$

ou seja, a distância mínima de pares d_P é pelo menos uma unidade maior que a distância mínima de Hamming d_H e pode chegar até o dobro.

Verificamos também que, apesar de um erro em um vetor de pares poder significar erro em uma ou em ambas coordenadas deste par, a capacidade de correção deste tipo de código não é prejudicada, ou seja, um código $\mathcal{C}$ pode corrigir até t pares de erros se, e somente se, $d_P(\mathcal{C}) \geq 2t + 1$, com $d_P(\mathcal{C})$ a distância mínima de pares do código $\mathcal{C}$.

Construimos dois tipos de códigos cíclicos cuja distância mínima de pares é pelo menos duas ou três unidades maior do que a distância de Hamming e verificamos que um código intercalado $\mathcal{C}$ possui distância mínima de pares exatamente $2d_H$, com d_H a distância mínima de $\mathcal{C}$. Além disso, um resultado ainda melhor para a distância de pares é demonstrado para códigos cíclicos lineares binários com dimensão maior do que 1, a saber,

$$d_P(\mathcal{C}) \geq d_H(\mathcal{C}) + \left\lceil \frac{d_H(\mathcal{C})}{2} \right\rceil.$$

O número de palavras de um código é um parâmetro importante e, assim como na Teoria Clássica, existem algumas cotas que exibimos neste trabalho ao final do Capítulo 2. Apresentamos limites inferiores e superiores nos tamanhos dos códigos de pares de símbolos e uma cota assíntotica que se mostrou estritamente maior que a cota assintótica de Gilbert-Varshamov para o mesmo número de erros cometidos. Definimos os códigos de pares MDS (*Maximum Distance Separable*) como apresentado em [5], por Chee *et al.* e apresentamos alguns códigos de pares que satisfazem esta definição.

A decodificação é uma etapa importante na transmissão de informações. No Capítulo 3 descrevemos três decodificadores para os códigos de pares. O primeiro proposto por Cassuto e Blaum [3] reduz o problema de decodificação de códigos de pares a um problema de decodificação de erros e exclusões na métrica de Hamming. O segundo, chamado de *algoritmo de decodificação por síndrome*, proposto em [14] por Hirotomo, Takita e Morii, utiliza o que definimos como *síndrome de pares* e *síndrome do símbolo vizinho* para criar um algoritmo de decodificação para códigos de pares. E o terceiro, proposto por Yaakobi, Bruck e Siegel em [29] e [28], usa decodificadores de códigos cíclicos na métrica

de Hamming para corrigir até t_0 pares de erros, com $t_0 = \left\lfloor \frac{3t+1}{2} \right\rfloor$ e $d_H(\mathcal{C}) = 2t + 1$. O primeiro e o terceiro algoritmos de decodificação possuem uma facilidade um pouco maior para cálculos que o segundo, pois reduzem a decodificação de erros de pares à decodificação na métrica de Hamming, que já é bem conhecida, entretanto não conseguem corrigir todos os pares de erros dentro da capacidade de correção de um código de pares. Já o segundo algoritmo possui capacidade de correção igual à capacidade do código.

No Capítulo 4 é feita uma extensão desta teoria para *códigos para canais de leitura de b-símbolos* como proposto por Yaakobi, Bruck e Siegel em [29] e [28], com $3 \leq b < n$. Neste caso a leitura é feita em blocos de b-símbolos. Apresentamos as definições básicas e dois resultados que comparam a b-distância entre duas palavra de um código binário com a distância de Hamming entre elas. O primeiro diz que se x é uma palavra de $\mathcal{A}^n$, com $0 < \omega_H(x) \leq n - (b-1)$, então

$$\omega_H(x) + b - 1 \leq \omega_b(x) \leq b \cdot \omega_H(x),$$

como ω_H o peso de Hamming e ω_b o b-peso do vetor. Já o segundo resultado diz que, para quaisquer $x \in \mathcal{A}^n$ e um inteiro positivo $b \geq 3$,

$$\omega_b(x) = \omega_H(\hat{x}) + (b-1) \cdot \frac{\omega_H(\hat{x}')}{2}.$$

Mostramos que a b-distância de um código intercalado $\mathcal{C}$ satisfaz $d_b(\mathcal{C}) = b \cdot d_H(\mathcal{C})$ e, ainda, descrevemos um algoritmo que corrige até $t = \left\lfloor \frac{d_b(\mathcal{C})-1}{2} \right\rfloor$ erros para este código. Além disto, também definimos o código de b-símbolos MDS (*Maximum Distance Separable*) como em [7] e fazemos um estudo da b-distância de duas classes de códigos com distância mínima de Hamming pequena, são elas o código "completo" $\mathcal{C} = \mathcal{A}^n$ e o código de Hamming linear cíclico.

No primeiro capítulo fazemos uma breve revisão de tópicos da Teoria Clássica dos Códigos Corretores de Erros e incluimos alguns resultados que utilizamos com frequência no desenvolvimento deste trabalho

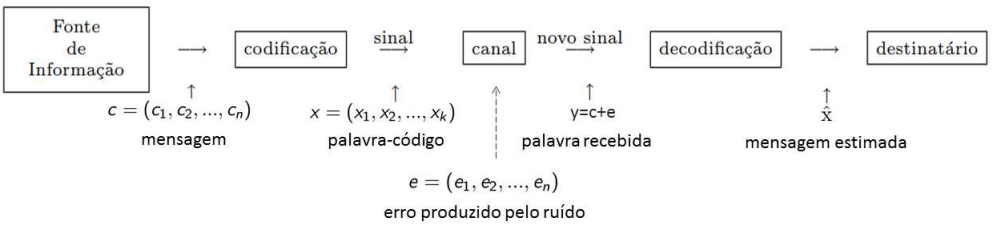
Capítulo 1

Códigos corretores de erros

Um artigo publicado pelo matemático C. E. Shannon em 1948, quando trabalhava nos Laboratórios Bell, marca o início da Teoria dos Códigos Corretores de Erros no artigo intitulado "*A Mathematical Theory of Communication*" [25].

Um exemplo prático da Teoria de Códigos Corretores de Erros no cotidiano atual é quando utilizamos um computador e digitamos uma palavra errada. Ele tenta corrigir essa palavra, nos indicando uma possibilidade de correção, que nem sempre será a palavra que queríamos digitar, mas pode ocorrer que o seja. Outros exemplos da Teoria de Códigos são as informações digitalizadas, como assistir televisão, falar ao telefone ou ouvir um CD.

A Figura 1 exemplifica como funciona a transmissão de informações.



Esquema básico de transmissão de informação.

Iniciamos este capítulo com tópicos da Teoria Clássica dos Códigos Corretores de Erros. Apresentamos algumas definições básicas, mostramos a capacidade de detecção e correção de erros de um código. Definimos o que é um código linear, verificamos algumas propriedades básicas e apresentamos um decodificador para códigos lineares que utiliza a *síndrome* de um vetor no processo de decodificação, o que será útil no Capítulo 3. Em seguida, definimos os códigos cíclicos e os códigos BCH e apresentamos algumas de suas

propriedades. Destacamos algumas cotas superiores e inferiores para o número de palavras de um código, as quais servem de base para determinar limitantes para este parâmetro também para os código de pares de símbolos, como veremos no final do Capítulo 2. Nas três últimas seções deste capítulo, apresentamos as definições e algumas propriedades da transformada discreta de Fourier, dos códigos concatenados e das funções piso e teto, que serão úteis no decorrer do trabalho. As principais referências para este capítulo são [1], [12], [13], [15], [19], [22] e [23]. Vamos denotar o vetor nulo como $\mathbf{0}$ e o vetor com todas as coordenadas 1 como $\mathbf{1}$.

1.1 Definições Básicas

Podemos dizer que um **código corretor de erros** é, em essência, um modo organizado de acrescentar algum dado adicional a cada informação que se queira transmitir ou armazenar, que permita, ao recuperar a informação, detectar e corrigir erros.

Os ingredientes necessários para a contrução de um código corretor de erros são:

- Um conjunto finito não vazio $\mathcal{A}$ que será chamado de **alfabeto**. O número de elementos de $\mathcal{A}$ será denotado por $q =\mid \mathcal{A} \mid$. Quando o alfabeto possui q elementos, dizemos que o código é *q-ário*. Quando o alfabeto é $\mathbb{Z}_2 = \{0, 1\}$, dizemos que o código é *binário*.

- Para n um natural não nulo, uma **palavra** v **de comprimento** n (ou vetor) escrita com o alfabeto $\mathcal{A}$ é uma sequência $v = (a_1, a_2, \ldots, a_n)$ (ou $v = a_1a_2 \cdots a_n$), com $a_i \in \mathcal{A}$, para todo i.

Definição 1.1.1. *Um* ***código corretor de erros*** q***-ário*** $\mathcal{C}$ *de comprimento* n *com palavras no alfabeto* $\mathcal{A}$ *é um subconjunto próprio de* $\mathcal{A}^n = \mathcal{A} \times \mathcal{A} \times \cdots \times \mathcal{A}$.

Definição 1.1.2. *Dados dois elementos* $u = (u_1, \ldots, u_n), v = (v_1, \ldots, v_n) \in \mathcal{A}^n$, *a* **distância de Hamming** *entre* u *e* v *é dada por*

$$d_H(u, v) = |\{i : u_i \neq v_i, 1 \leq i \leq n\}|.$$

As demonstrações dos resultados apresentados a seguir são omitidos mas podem ser encontrados em [13].

Proposição 1.1.3. *Dados* $u, v, w \in \mathcal{A}$, *valem as seguintes propriedades:*

(i) Positividade: $d_H(u, v) \geq 0$, *valendo a igualdade se, e somente se,* $u = v$.

(ii) Simetria: $d_H(u, v) = d_H(v, u)$.

(iii) Desigualdade Triangular: $d_H(u,v) \leq d_H(u,w) + d_H(w,v)$.

Esse resultado mostra que a distância de Hamming é uma métrica e, por esse motivo, a distância de Hamming também é chamada de *métrica de Hamming*.

Definição 1.1.4. *Seja* $\mathcal{C} \subset \mathcal{A}^n$ *um código. A* **distância mínima** *de* $\mathcal{C}$ *é o número*

$$d_H = d_H(\mathcal{C}) = min\{d_H(u,v);\ u,v \in \mathcal{C}\, e\, u \neq v\}.$$

Um código corretor de erros com comprimento n, número de palavras M e distância mínima d_H é chamado de (n, M, d_H)**-código**.

Definição 1.1.5. *Dados* $a \in \mathcal{A}^n$ *e* $t > 0$ *um número real, define-se* ***disco*** *e* ***esfera*** *de centro* a *e raio* t*, respectivamente, por:*

$$D(a,t) = \{u \in \mathcal{A}^n;\ d_H(u,a) \leq t\} \quad e \quad S(a,t) = \{u \in A^n;\ d_H(u,a) = t\}.$$

Lema 1.1.6. *Para todo* $a \in \mathcal{A}^n$ *e todo número real* $r > 0$*, temos*

$$|D(a,r)| = \sum_{i=0}^{r} \binom{n}{i} (q-1)^i.$$

Lema 1.1.7. *Seja* $\mathcal{C}$ *um código com distância mínima* d_H*. Se* c *e* c' *são palavras distintas de* $\mathcal{C}$*, então*

$$D(c,\kappa) \bigcap D(c,\kappa) = \emptyset,$$

com $\kappa = \left\lfloor \frac{d_H - 1}{2} \right\rfloor$.

Teorema 1.1.8. *Seja* $\mathcal{C}$ *um código* q*-ário de comprimento* n *com distância mínima* d_H*. Então* $\mathcal{C}$ *pode detectar até* $d_H - 1$ *erros e corrigir até* $\kappa = \left\lfloor \frac{d_H - 1}{2} \right\rfloor$ *erros.*

Uma consequência do Teorema 1.1.8 é que um código terá capacidade de correção de erros maior quanto maior for a sua distância mínima.

Definição 1.1.9. *Sejam* $\mathcal{C} \subset \mathcal{A}^n$ *um código* q*-ário com distância mínima* d_H *e* $\kappa = \left\lfloor \frac{d_H - 1}{2} \right\rfloor$*. O código* $\mathcal{C}$ *será dito* **perfeito** *se*

$$\bigcup_{c \in \mathcal{C}} D(c,\kappa) = \mathcal{A}^n.$$

1.2 Códigos Lineares

A classe de códigos mais utilizada na prática são os *códigos lineares*, para os quais o alfabeto é $\mathbb{F}_q$, um **corpo finito** com q elementos. Para cada número natural n, $\mathbb{F}_q^n$ é um

$\mathbb{F}_q$-espaço vetorial de dimensão n. As demonstrações dos resultados desta seção podem ser encontrados em [13].

Definição 1.2.1. *Um código q-ário $\mathcal{C} \subset \mathbb{F}_q^n$ será chamado* ***código linear*** *se for um subespaço vetorial próprio de $\mathbb{F}_q^n$.*

Se $\mathcal{C}$ tem dimensão k sobre $\mathbb{F}_q$ e distância mínima d_H, então diremos que $\mathcal{C}$ é um (n, k, d_H)**-código linear**.

Seja k a dimensão de um código $\mathcal{C}$ e seja $\{\mathbf{v_1}, ..., \mathbf{v}_k\}$ uma base. Todo elemento de $\mathcal{C}$ se escreve como combinação linear, de modo único, na forma

$$\lambda_1\mathbf{v_1} + \lambda_\mathbf{2}\mathbf{v_2} + ... + \lambda_k\mathbf{v}_k,$$

com $\lambda_i \in \mathbb{F}_q$, para todo $i = 1, ..., k$. Daí

$$M = |\mathcal{C}| = q^k,$$

e, conseqüentemente, $dim_{\mathbb{K}}\mathcal{C} = k = log_q q^k = log_q M$.

Definição 1.2.2. *Dado $x \in \mathbb{F}_q^n$, define-se o* ***peso*** *de x como sendo o número inteiro*

$$\omega_H(x) = |\{i;\ x_i \neq 0\}|.$$

Em outras palavras, $\omega_H(x) = d_H(x, 0)$, com d_H a métrica de Hamming.

Definição 1.2.3. *O* ***peso*** *de um código linear $\mathcal{C}$ é o inteiro*

$$\omega_H(\mathcal{C}) = min\{\omega_H(x);\ x \in \mathcal{C}\backslash\{\mathbf{0}\}\}.$$

Proposição 1.2.4. *Seja $\mathcal{C} \subset \mathbb{F}_q^n$ um código linear com distância mínima d_H. Temos:*

(i) para quaisquer $x, y \in \mathbb{F}_q^n$, $d_H(x, y) = \omega_H(x - y)$.

(ii) $d_H = \omega_H(\mathcal{C})$.

Sejam $\mathbb{F}_q$ um corpo finito com q elementos e $\mathcal{C} \subset \mathbb{F}_q^n$ um código linear. Seja $\beta = \{v_1, ..., v_k\}$ uma base ordenada de $\mathcal{C}$ e considere a matriz G, cujas linhas são os vetores $v_i = (b_{i1}, \ldots, b_{in})$, $i = 1, \ldots, k$, isto é,

$$G = \begin{pmatrix} b_{11} & b_{12} & \cdots & b_{1n} \\ \vdots & \vdots & & \vdots \\ b_{k1} & b_{k2} & \cdots & b_{kn} \end{pmatrix}.$$

A matriz G é chamada de **matriz geradora** de $\mathcal{C}$ associada à base β. Note que $\mathcal{C}$ é o subespaço gerado pelas linhas de G, assim, os elementos de $\mathcal{C}$ são todas as palavras $y \in \mathbb{F}_q^n$ tais que $xG = y$, para $x \in \mathbb{F}_q^k$.

Lembramos que G não é univocamente determinada, pois depende da escolha da base β de $\mathcal{C}$.

Definição 1.2.5. *Uma matriz geradora G de um código $\mathcal{C}$ está na **forma padrão** se*

$$G = (Id_k \mid A),$$

com Id_k a matriz identidade de ordem k e A uma matriz $k \times (n-k)$.

Seja $\mathcal{C} \subset \mathbb{F}_q^n$ um código linear. Define-se o **conjunto ortogonal a $\mathcal{C}$** em $\mathbb{F}_q^n$ por

$$\mathcal{C}^{\perp} = \{v \in \mathbb{F}_q^n;\ \langle u, v\rangle = 0,\ \text{para todos } u \in \mathcal{C}\}.$$

Sendo $\langle ., .\rangle$ o produto interno sobre $\mathbb{F}_q$.

Lema 1.2.6. *Se $\mathcal{C} \subset \mathbb{F}_q^n$ é um código linear, com matriz geradora G, então*

(i) $\mathcal{C}^{\perp}$ é um subespaço vetorial próprio de $\mathbb{F}_q^n$.

(ii) $x \in \mathcal{C}^{\perp}$ se, e somente se, $Gx^T = 0$.

Definição 1.2.7. *O subespaço vetorial $\mathcal{C}^{\perp}$ de $\mathbb{F}_q^n$ é também um código linear e o chamaremos **código dual** de $\mathcal{C}$.*

Sejam $\mathcal{C}$ um (n,k)-código linear com matriz de codificação G e $y \in \mathcal{C}^{\perp}$. Então

$$\langle x, y\rangle = 0, \text{ para todo } x \in \mathcal{C}.$$

Logo y é ortogonal a todos os elementos de uma base de $\mathcal{C}$, o que equivale a $G \cdot y^T = 0$, já que a matriz G é formada pelos vetores de uma base de $\mathcal{C}$ em suas linhas.

Assim, podemos definir o código dual $\mathcal{C}^{\perp}$ da seguinte forma

$$\mathcal{C}^{\perp} = \{y \in \mathbb{F}_q^n;\ G \cdot y^T = 0\}.$$

Proposição 1.2.8. *Seja $\mathcal{C} \subset \mathbb{F}_q^n$ um código de dimensão $k < n$ com matriz geradora $G = (Id_k \mid A)$ na forma padrão. Então*

(i) $dim\, \mathcal{C}^{\perp} = n - k$,

(ii) $H = (-A^T | Id_{n-k})$ *é uma matriz geradora de* $\mathcal{C}^\perp$ *e*

(iii) $(\mathcal{C}^\perp)^\perp = \mathcal{C}$.

Para identificar de um vetor $v \in \mathbb{F}_q^n$ pertence ou não a um código $\mathcal{C} \subset \mathbb{F}_q^n$ basta utilizar o seguinte resultado.

Proposição 1.2.9. *Seja* $\mathcal{C} \subset \mathbb{F}_q^n$ *um código linear tal que* $\mathcal{C}^\perp$ *tem matriz geradora* H. *Então*

$$v \in \mathcal{C} \iff Hv^T = 0.$$

A matriz H geradora de $\mathcal{C}^\perp$ é chamada **matriz teste de paridade** de $\mathcal{C}$.

Definição 1.2.10. *Dados um código linear* $\mathcal{C} \subset \mathbb{F}_q^n$ *com matriz teste de paridade* H *e um vetor* $v \in \mathbb{F}_q^n$, *dizemos que* Hv^T *é a* **síndrome** *de* v.

A Proposição 1.2.11 e o Teorema 1.2.12 apresentados a seguir fornecem uma relação entre a matriz teste de paridade e peso mínimo de Hamming d_H de um código.

Proposição 1.2.11. *Dado um código* $\mathcal{C} \subset \mathbb{F}_q^n$ *com matriz teste de paridade* H, *o peso de* $\mathcal{C}$ *é maior ou igual a* s *se, e somente se, quaisquer* $s-1$ *colunas de* H *são linearmente independentes.*

Teorema 1.2.12. *Seja* H *a matriz teste de paridade de um código* $\mathcal{C}$. *Então* $\omega_H(\mathcal{C}) = s$ *se, e somente se, quaisquer* $s-1$ *colunas de* H *são linearmente independentes e existem* s *colunas de* H *linearmente dependentes.*

No Capítulo 3 vamos abordar alguns algoritmos de decodificação para códigos de pares de símbolos, um deles é um algoritmo de decodificação por síndrome, análogo ao algoritmo de decodificação para códigos clássicos que veremos a seguir.

1.3 Decodificação

Uma etapa importante da transmissão de informações é a detecção e correção de erros. Chama-se **decodificação** este processo de detecção e correção de erros em um determinado código. Nesta seção vamos apresentar o algoritmo de decodificação por síndrome, as demonstrações dos resultados serão omitidos e podem ser encontrados em [13].

Suponha que um vetor c transmitido sofreu algum tipo de interferência e foi recebido como outro vetor r. O **vetor erro** e é definido como a diferença entre o vetor recebido r e o transmitido c, ou seja,

$$e = r - c.$$

O peso do vetor erro determina quantos erros foram cometidos desde a transmissão até a recepção.

Se H é a matriz teste de paridade de um código $\mathcal{C}$, então $Hc^T = 0$, para todo $c \in \mathcal{C}$. Com isso,

$$He^T = H(r^T - c^T) = Hr^T - Hc^T = Hr^T.$$

Logo e e r têm a mesma síndrome.

Para melhor entendimento do lema seguinte, denotemos por h^i a i-ésima coluna de H. Se $e = (\alpha_1, \ldots, \alpha_n)$, então

$$\sum_{i=1}^{n} \alpha_i h^i = He^T = Hr^T.$$

Lema 1.3.1. *Seja $\mathcal{C} \subset \mathbb{F}_q^n$ um código linear com capacidade de correção κ. Se $r \in \mathbb{F}_q^n$ e $c \in \mathcal{C}$ são tais que $d_H(c, r) \leq \kappa$, então existe um único vetor e com $\omega_H(e) \leq \kappa$, cuja síndrome é igual a síndrome de r e $c = r - e$.*

Seja $\mathcal{C} \subset \mathbb{F}_q^n$ um código corretor de erros com matriz teste de paridade H, distância mínima d_H e $\kappa = \left\lfloor \dfrac{d_H - 1}{2} \right\rfloor$. Seja $v \in \mathbb{F}_q^n$ e defina

$$v + \mathcal{C} = \{v + c;\ c \in \mathcal{C}\}$$

como a **classe lateral** de v segundo $\mathcal{C}$.

Lema 1.3.2. *Os vetores u e v de $\mathbb{F}_q^n$ têm a mesma síndrome se, e somente se, $u \in v + \mathcal{C}$.*

A proposição a seguir nos diz que as diferentes classes laterais de $\mathcal{C}$ em $\mathbb{F}_q^n$ formam uma partição de $\mathbb{F}_q^n$, cujas partes têm a mesma cardinalidade.

Proposição 1.3.3. *Seja $\mathcal{C} \subset \mathbb{F}_q^n$ um código linear de dimensão k. Temos:*

(i) $v + \mathcal{C} = v' + \mathcal{C} \quad \Leftrightarrow \quad v - v' \in \mathcal{C}$;

(ii) $(v + \mathcal{C}) \cap (v' + \mathcal{C}) \neq \emptyset \quad \Longrightarrow \quad v + \mathcal{C} = v' + \mathcal{C}$;

(iii) $\bigcup_{v \in \mathbb{F}_q^n} (v + \mathcal{C}) = \mathbb{F}_q^n$;

(iv) $|(v+\mathcal{C})| = |\mathcal{C}| = q^k$.

De (ii) e (iv) da Proposição 1.3.3, segue que o número de classes laterais segundo $\mathcal{C}$ é

$$\frac{q^n}{q^k} = q^{n-k}.$$

Definição 1.3.4. *Um vetor de peso mínimo numa classe lateral é chamado de* ***elemento líder*** *dessa classe.*

Proposição 1.3.5. *Seja $\mathcal{C}$ um código linear em $\mathbb{F}_q^n$ com distância mínima d_H. Se $u \in \mathbb{F}_q^n$ é tal que*

$$\omega_H(u) \leq \left\lfloor \frac{d_H - 1}{2} \right\rfloor = \kappa,$$

então u é o único elemento líder de sua classe.

Esta proposição fornece informações importantes para a procura de elementos líderes de classes, pois basta tomar aqueles elementos cujo peso é menor ou igual a κ.

Vamos agora apresentar um algoritmo de correção de mensagens que tenham sofrido um número de erros menor ou igual à capacidade $\kappa = \left\lfloor \frac{d_H-1}{2} \right\rfloor$ de correção do código.

Preparação: Determine todos os elementos u de $\mathbb{F}_q^n$ tais que $\omega_H(u) \leq \kappa$. Em seguida calcule suas síndromes e coloque esses dados em uma tabela. Seja $\mathbf{r}$ uma palavra recebida.

Algoritmo de Decodificação por Síndrome

1. Calcule a síndrome $H\mathbf{r}^{\mathbf{T}} = \mathbf{s}^{\mathbf{T}}$.
2. Se $\mathbf{s}$ está na tabela, seja l o elemento líder da classe determinada por $\mathbf{s}$; troque $\mathbf{r}$ por $\mathbf{r}$ - l.
3. Se $\mathbf{s}$ não está na tabela, então foram cometidos mais do que κ erros no envio dessa mensagem e a palavra não pode ser corrigida.

Justificativa: Dado r, sejam c e e, respectivamente, a mensagem transmitida e o vetor erro. Como o vetor erro e a mensagem recebida têm a mesma síndrome, então a classe lateral na qual e se encontra está determinada pela síndrome de r. Se $\omega_H(e) \leq \kappa$, então e é o único elemento líder l de sua classe e, portanto, é conhecido e se encontra na tabela. Conseqüentemente, pelo Lema 1.3.1, $c = r - e = r - l$ é determinado.

Exemplo 1.3.6. *Considere o* $(6,3)$*-código linear definido sobre* $\mathbb{F}_2$ *com matriz teste de paridade*

$$H = \begin{pmatrix} 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 \end{pmatrix}.$$

Neste caso, $d_H = 3$ *e* $\kappa = \left[\frac{d_H - 1}{2}\right] = 1$*. Os vetores de peso* ≤ 1*, com as suas respectivas síndromes estão relacionados na tabela a seguir:*

líder	*síndrome*
000000	*000*
000001	*101*
000010	*011*
000100	*110*
001000	*001*
010000	*010*
100000	*100*

Suponhamos agora que a palavra recebida seja

(a) $r = (100011)$*. Logo* $Hr^T = (010)^T$ *e, portanto,* $e = (010000)$*. Consequentemente,* $c = r - e = (110011)$*.*

(b) $r = (111111)$*. Logo* $Hr^T = (111)^T$*, que não se encontra na tabela. Com isso, foi cometido mais do que 1 erro na mensagem r.*

1.4 Códigos Cíclicos

Os códigos cíclicos são uma importante subclasse dos códigos lineares pois possuem bons algoritmos de codificação e decodificação.

Seja $\mathbb{F}_q$ um corpo finito. Representamos os vetores $\mathbb{F}_q^n$ por $(x_0, x_1, \ldots, x_{n-1})$.

Definição 1.4.1. *Um código linear* $\mathcal{C} \subset \mathbb{F}_q^n$ *é um* ***código cíclico*** *se, para todo* $\boldsymbol{c} = (c_0, \ldots, c_{n-1})$ *pertencente a* $\mathcal{C}$*, o vetor* $(c_{n-1}, c_0, \ldots, c_{n-2}) \in \mathcal{C}$*.*

Defina a permutação π de $\{0, 1, \ldots, n-1\}$ por

$$\pi(i) = \begin{cases} i - 1, & se \quad i \geq 1 \\ n - 1, & se \quad i = 0 \end{cases}.$$

Assim, $T_\pi(c_0, \ldots, c_{n-1}) = (c_{n-1}, c_0, \ldots, c_{n-2})$, para todo $(c_0, c_1, \ldots, c_{n-1}) \in \mathbb{F}_q^n$. Portanto, e equivalentemente, dizemos que um código $\mathcal{C}$ é cíclico se, e somente se, $T_\pi(\mathcal{C}) \subset \mathcal{C}$.

Exemplo 1.4.2.

1. $\mathcal{C} = \{(00000, 10100, 01010, 00101, 10010, 01001)\}$ *é um código cíclico.*

2. *Dado* $v \in \mathbb{F}_q^n$*, o subespaço vetorial de* $\mathbb{F}_q^n$

$$\langle v \rangle = \mathbb{F}_q v + \mathbb{F}_q T_\pi v + \cdots + \mathbb{F}_q T_\pi^{n-1} v$$

é um código cíclico. De fato, temos

$$T_\pi(\mathbb{F}_q v + \mathbb{F}_q T_\pi v + \cdots \mathbb{F}_q T_\pi^{n-1} v) = \mathbb{F}_q T_\pi(v) + \mathbb{F}_q T_\pi^2(v) + \cdots + \mathbb{F}_q T_\pi^{n-1}(v) + \mathbb{F}_q v$$

e assim $T_\pi(\langle v \rangle) \subset \langle v \rangle$.

A seguir vamos caracterizar os códigos cíclicos como ideais de um anel. As demonstrações dos resultados apresentados a seguir são omitidos mas podem ser encontrados em [13].

Considere R_n o **anel das classe residuais** em $\mathbb{F}_q[x]$ módulo $x^n - 1$, ou seja,

$$R_n = \frac{\mathbb{F}_q[x]}{\langle x^n - 1 \rangle}.$$

Um elemento de R_n é um conjunto da forma

$$[f(x)] = \{f(x) + g(x)(x^n - 1);\ g(x) \in \mathbb{F}_q[x]\};$$

e a adição e multiplicação em R_n são definidas por

$$[f_1(x)] + [f_2(x)] = [f_1(x) + f_2(x)] \tag{1.1}$$

e por

$$[f_1(x)] \cdot [f_2(x)] = [f_1(x) \cdot f_2(x)],$$

respectivamente. Recorde também que R_n munido da adição (1.1) e multiplicação por escalares $\lambda \in \mathbb{F}_q$, definida por

$$\lambda[f(x)] = [\lambda f(x)],$$

é um $\mathbb{F}_q$-espaço vetorial de dimensão n, com base $1, [x], \ldots, [x^{n-1}]$ e, assim, é isomorfo a

$\mathbb{F}_q^n$ pelo isomorfismo

$$\begin{array}{rccc} \nu: & \mathbb{F}_q^n & \longrightarrow & R_n \\ & (c_0,\ldots,c_{n-1}) & \mapsto & [c_0+c_1x+\cdots+c_{n-1}x^{n-1}]. \end{array}$$

Aplicar T_π em $\mathbb{F}_q^n$ se traduz na multiplicação por $[x]$ em R_n, pois

$$\nu(T_\pi(\mathbf{c})) = [c_{n-1}+c_0x+\cdots+c_{n-2}x^{n-1}] = [x][c_0+c_1x+\cdots+c_{n-1}x^{n-1}] = [x]\nu(\mathbf{c})\ .$$

As demonstrações do Lema 1.4.3 e do Teorema 1.4.4 decorrem diretamente destas observações acima.

Lema 1.4.3. *Seja V um subespaço vetorial de R_n. Então V é um ideal de R_n se, e somente se, V é fechado para a multiplicação por $[x]$.*

Teorema 1.4.4. *Um subespaço vetorial $\mathcal{C} \subset \mathbb{F}_q^n$ é um código cíclico se, e somente se, $\nu(\mathcal{C})$ é um ideal de R_n.*

Os resultados a seguir são consequências do fato do anel $\mathbb{F}_q[x]$ ser um domínio de ideais principais.

Proposição 1.4.5. *Todo ideal de $\mathbb{F}_q[x]$ é da forma $I(f(x)) = \{h(x)f(x);\ h(x) \in \mathbb{F}_q[x]\}$, para algum $f(x) \in \mathbb{F}_q[x]$.*

Corolário 1.4.6. *Seja $I \neq \{0\}$ um ideal de $\mathbb{F}_q[x]$. Então existe um único polinômio mônico em $g(x) \in I$ (de grau mínimo) tal que $I = (g(x))$.*

Proposição 1.4.7. *Todo ideal de R_n é da forma $I([f(x)])$, com $f(x)$ um divisor mônico de x^n-1.*

Assim, do Teorema 1.4.4 e da Proposição 1.4.7 temos que um código $\mathcal{C}$ de $\mathbb{F}_q^n$ é um código cíclico se, e somente se, $\nu(\mathcal{C})$ é da forma $I([f(x)])$, com $f(x)$ um divisor de x^n-1. O polinômio $f(x)$ é dito **polinômio gerador** de $\mathcal{C}$. Portanto, conhecendo a fatoração de x^n-1, conhecemos todos os códigos cíclicos de $\mathbb{F}_q^n$.

Vamos agora encontrar matrizes geradoras e matrizes teste de paridade para um código cíclico caracterizado por um polinômio gerador.

Teorema 1.4.8. *Seja $I = I([g(x)])$, com $g(x)$ um divisor de x^n-1 de grau s. Então*

$$B = \{[g(x)], [xg(x)], [x^2g(x)], \ldots, [x^{n-s-1}g(x)]\}$$

é uma base de I como espaço vetorial sobre $\mathbb{F}_q$.

Corolário 1.4.9. *Dado um código cíclico $\mathcal{C}$, existe $v \in \mathcal{C}$ tal que $\mathcal{C} = \langle v \rangle$.*

Corolário 1.4.10. *Seja $g(x) = g_0 + g_1x + \cdots + g_sx^s$ um divisor de $x^n - 1$ de grau s. Se $I = I([g(x)])$, então*

$$\dim_{\mathbb{F}_q} I = n - s \ ,$$

e o código $\mathcal{C} = \nu^{-1}(I)$ tem matriz geradora

$$G = \begin{pmatrix} \nu^{-1}([g(x)]) \\ \nu^{-1}([xg(x)]) \\ \vdots \\ \nu^{-1}([x^{n-s-1}g(x)]) \end{pmatrix} = \begin{pmatrix} g_0 & g_1 & \cdots & g_s & 0 & \cdots & 0 \\ 0 & g_0 & g_1 & \cdots & g_s & \cdots & 0 \\ \vdots & \vdots & \vdots & & \vdots & & \vdots \\ 0 & \cdots & 0 & g_0 & \cdots & & g_s \end{pmatrix}$$

Definição 1.4.11. *Dado o polinômio $h(x) = h_0 + h_1x + \cdots + h_tx^t$ que divide $x^n - 1$, o **polinômio recíproco** de $h(x)$, dado por*

$$h^*(x) = x^t h(x^{-1}) \ ,$$

também divide $x^n - 1$ e é, portanto, gerador de algum código cíclico em $\mathbb{F}_q^n$.

Teorema 1.4.12. *Seja $\mathcal{C}$ um código cíclico, com $I = I([g(x)])$ e*

$$g(x) = g_0 + g_1x + \cdots + g_sx^s$$

um divisor de $x^n - 1$ de grau s tal que $x^n - 1 = g(x)h(x)$, com

$$h(x) = h_0 + h_1x + \cdots + h_{n-s-1}x^{n-s-1}.$$

Então $\mathcal{C}^\perp$ é cíclico e $\mathcal{C}^\perp = \nu^{-1}(J)$, com $J = I([h^*(x)])$.

Como conseqüência imediata temos o seguinte resultado:

Corolário 1.4.13. *Uma matriz teste de paridade de $\mathcal{C} = \nu^{-1}(I)$, em que $I = I([g(x)])$ é dada por*

$$\begin{pmatrix} h_{n-s} & h_{n-s-1} & \cdots & h_0 & 0 & \cdots & 0 \\ 0 & h_{n-s} & h_{n-s-1} & \cdots & h_0 & \cdots & 0 \\ \vdots & \vdots & \vdots & & \vdots & & \vdots \\ 0 & \cdots & 0 & h_{n-s} & \cdots & & h_0 \end{pmatrix},$$

com

$$\frac{x^n - 1}{g(x)} = h_0 + h_1x + \cdots + h_{n-s}x^{n-s} \ .$$

1.5 Códigos de Hamming

Um exemplo de código muito comum são os códigos de Hamming, eles são uma família de códigos importante pois são fáceis de codificar e decodificar além de serem códigos perfeitos.

Definição 1.5.1 (Códigos de Hamming)**.** *Um* ***código de Hamming*** *de ordem m sobre $\mathbb{F}_2$ (corpo com 2 elementos) é um código $\mathcal{C}$ com matriz teste de paridade H_m, de ordem $m \times n$, cujas colunas são os elementos de $\mathbb{F}_2^m \backslash \{\mathbf{0}\}$ numa ordem qualquer.*

Assim, seja $\mathcal{C} \subset \mathbb{F}_2^n$ o código determinado pela matriz H_m. Temos $n = 2^m - 1$, pela própria construção de H_m. Com isso, sua dimensão é $k = n - m = 2^m - m - 1$. É fácil observar que a distância mínima de Hamming de um código de Hamming é $d_H = 3$.

Exemplo 1.5.2. *Considere a matriz*

$$H_3 = \begin{pmatrix} 1 & 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 1 \end{pmatrix}.$$

Esta é a matriz de um código de Hamming correspondente a $m = 3$. Note que o comprimento deste código é $n = 2^3 - 1 = 7$ e sua dimensão é $k = 7 - 3 = 4$.

A demonstração da proposição a seguir pode ser encontrada em [13].

Proposição 1.5.3. *Todo código de Hamming é perfeito.*

Existe outra forma de definirmos os códigos de Hamming, com uma ordem determinada na disposição dos vetores nas colunas da matriz H_m. Neste caso, diremos que o código de Hamming é cíclico. Para isto precisamos da seguinte definição.

Definição 1.5.4. *Um elemento α de um corpo finito $\mathbb{F}_q$ com q elementos é chamado de* **elemento primitivo** *se*

$$\mathbb{F}_q^* = \{1, \alpha, \alpha^2, \ldots, \alpha^{q-2}\},$$

ou seja, se $\operatorname{ord} \alpha = q - 1$. *Com* $\operatorname{ord} \alpha$ *a ordem do elemento $\alpha \in \mathbb{F}_q^*$.*

Como visto acima, os códigos de Hamming de ordem m possuem matriz teste de paridade cujas colunas são todos os elementos de $\mathbb{F}_2^m \setminus \{\mathbf{0}\}$. Agora, se α é um elemento primitivo de $\mathbb{F}_q$, um corpo com p^m elementos, então $1, \alpha, \alpha^2, \ldots, \alpha^{2m-2}$ são distintos e podem ser representados por distintos elementos de $\mathbb{F}_2^m$. Assim, os códigos binários de

Hamming H_3 com parâmetros $(n = 2^m - 1, k = n - m, d_H = 3)$ tem uma matriz teste de paridade que pode ser tomada como

$$H = (1, \alpha, \alpha^2, \ldots, \alpha^{2m-2}),$$

em que cada entrada será substituída por um vetor coluna com m $0'$s e $1'$s.

O código de Hamming binário definido desta forma é um código cíclico (Cap. 7, [19]).

Exemplo 1.5.5. *Para H_3,*

$$\begin{aligned} H &= (1, \alpha, \alpha^2, \alpha^3, \alpha^4, \alpha^5, \alpha^6) \\ &= \begin{pmatrix} 0 & 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 1 & 1 \end{pmatrix}, \end{aligned}$$

com $\alpha \in \mathbb{F}_{2^3}$ satisfazendo $\alpha^3 + \alpha + 1 = 0$.

A seguir definimos os *códigos cíclicos encurtados*, que serão utilizados no Exemplo 2.3.1, como descritos em [[19], Cap. 7].

Definição 1.5.6 (Códigos cíclicos encurtados)**.** *Forme um conjunto com todas as palavras de um código cíclico $\mathcal{C}$ de comprimento n e distância mínima d_H que começam com i zeros consecutivos e delete estes zeros. O código resultante $\mathcal{C}^*$ tem comprimento $n - i$ e não é cíclico, entretanto, existe um polinômio $f(x)$ tal que $\mathcal{C}^*$ é um ideal no anel de polinômios* $\bmod f(x)$*, e reciprocamente, qualquer ideal neste anel é um* **código cíclico encurtado***.*

1.6 Códigos BCH

Na Seção 1.4 vimos que os códigos cíclicos podem ser vistos como ideais numa álgebra de polinômios em uma indeterminada, o que pode facilitar alguns cálculos. Entretanto, determinar a distância mínima desses códigos não é uma tarefa fácil. Veremos nesta seção uma família de códigos cíclicos, neste contexto polinômial, que possuem cotas inferiores para a distância mínima.

Definição 1.6.1. *Seja $\mathbb{F}_q$ um corpo finito, K um subcorpo de $\mathbb{F}_q$ e $\beta \in \mathbb{F}_q$. O conjunto*

$$J_\beta = \{p(x) \in K[x];\ p(\beta) = 0\} \neq \{0\}$$

é um ideal de $K[x]$, logo, J_β é da forma $I(m_\beta(x))$ para um único polinômio mônico $m_\beta(x) \in K[x]$. Esse polinômio é chamado de ***polinômio mínimo*** *ou* ***polinômio minimal*** *de β sobre K, e é um polinômio irredutível de menor grau em $K[x]$ tal que $p(\beta) = 0$.*

Sejam L e $\mathbb{F}_q$ corpos finitos tais que L é uma extensão de $\mathbb{F}_q$ e seja $\beta \in L$. O conjunto

$$\mathbb{F}_q(\beta) = \left\{ \frac{p(\beta)}{q(\beta)};\ p(x), q(x) \in \mathbb{F}_q[x],\ q(\beta) \neq 0 \right\}$$

é um subcorpo de L contendo β e $\mathbb{F}_q$ e contido em qualquer subcorpo de L que tem essa propriedade.

As demonstrações dos resultados a seguir são omitidas e podem ser encontradas em [13].

Proposição 1.6.2. *Sejam $\mathbb{F}_q$ um corpo finito, L um subcorpo de $\mathbb{F}_q$ e $\beta \in \mathbb{F}_q$.*

(i) Se $m = gr(m_\beta(x))$, então $1, \beta, \ldots, \beta^{m-1}$ é uma base de $L(\beta)$ sobre L. Em particular, $dim_L L(\beta) = gr(m_\beta(x))$.

(ii) Se $q = \mid L \mid$, então $\beta^{q^m} = \beta$ e $\beta^{q^i} \neq \beta^{q^j}$ para $i \neq j$, $i, j = 0, \ldots, m-1$. Além disso,

$$m_\beta(x) = (x - \beta)(x - \beta^q) \ldots (x - \beta^{q^{m-1}}).$$

(iii) Existe $\alpha \in \mathbb{F}_q$ tal que $\mathbb{F}_q = L(\alpha)$.

Seja $\mathcal{C} \subset \mathbb{F}_q^n$ um código cíclico com n e q primos entre si. Da Seção 1.4, o código $\mathcal{C}$ pode ser considerado com um ideal $\nu(C) = I([g(x)])$ no anel $R_n = \frac{\mathbb{F}_q[x]}{<x^n - 1>}$, com $g(x) \in \mathbb{F}_q[x]$ um divisor de $x^n - 1$.

Proposição 1.6.3. *Seja K um corpo finito que contém um corpo $\mathbb{F}_q$ sobre o qual o polinômio $x^n - 1$ se fatora em fatores lineares mônicos distintos. Sejam $\alpha_1, \ldots, \alpha_r$ as raízes de $g(x)$ em K, que são duas a duas distintas. Então*

$$\nu(C) = I([g(x)]) = \{[f(x)] \in R_n;\ f(\alpha_1) = \ldots = f(\alpha_r) = 0\}.$$

No resultado a seguir observa-se que, construindo um código cíclico com uma conjunto específico de raízes n-ésimas da unidade, temos uma cota inferior para a sua distância mínima de Hamming. Esses códigos são chamados **códigos BCH**.

Teorema 1.6.4 (Bose-Chaudhuri-Hocquenguem)**.** *Seja $\mathbb{F}_q$ um corpo com q elementos e n um inteiro maior do que 1 e primo com q. Seja $\mathbb{F}$ um corpo no qual $x^n - 1$ se decompõe em fatores lineares e seja $\gamma \in \mathbb{F}$ uma raiz n-ésima primitiva da unidade. Seja $\mathcal{C}$ um código cíclico com polinômio gerador $g(x) = \text{mmc}(m_{\gamma^a}(x), \ldots, m_{\gamma^{a+\delta-2}}(x))$, com $a \geq 0$ e $\delta \leq n$. Então a distância mínima de $\mathcal{C}$ é pelo menos δ e sua dimensão é pelo menos $n - m(\delta - 1)$, com $m = dim_{\mathbb{F}_q}\mathbb{F}$.*

O número δ que aparece no enunciado do Teorema é chamado **peso estimado** do código BCH, pois representa uma estimativa para a distância mínima do código.

Fixamos um corpo $\mathbb{F}_q$ e uma extensão $\mathbb{F}$ com uma raiz n-ésima primitiva da unidade $\gamma \in \mathbb{F}$ definimos

$$\mathcal{C}_{\mathbb{F}_q}(n,\delta) = \left\{ (a_0,\ldots,a_{n-1}) \in \mathbb{F}_q^n; \sum_{i=0}^{n-1} a_i \gamma^{ij} = 0, j = 1,\ldots,\delta-1 \right\},$$

ou seja, o código BCH definido pelo polinômio gerador $g(x) = \text{mmc}(m_\gamma(x),\ldots,m_{\gamma^{\delta-1}}(x))$.

Definição 1.6.5. *Um código BCH em $\mathbb{F}_q^n$, com $\mathbb{F}_q$ um corpo com q elementos e $n = q^m - 1$, para algum m, será chamado de* **código primitivo**.

Para códigos primitivos, sempre ocorre que n e $car(\mathbb{F}_q)$ são primos entre si, pois $\text{mdc}(q, q^m - 1) = 1$. Com $car(\mathbb{F}_q)$ a característica do corpo $\mathbb{F}_q$.

Proposição 1.6.6. *Seja $\mathcal{C} = \mathcal{C}_{\mathbb{F}_q}(n,\delta)$ um código BCH, com $\mathbb{F}_q$ um corpo com q elementos. Suponha $n = q^m - 1$, para algum inteiro m, e $\delta = q^h - 1$, para algum inteiro h com $h < m$. Então $\mathcal{C}$ tem peso $d_H = \delta$.*

Teorema 1.6.7. *Seja $\mathcal{C} = \mathcal{C}_{\mathbb{F}_q}(n,\delta)$ um código BCH primitivo, com $\mathbb{F}_q$ um corpo com q elementos. Então $\mathcal{C}$ tem peso $d_H \leq q\delta - 1$.*

Códigos BCH Generalizados

Hartman e Tzeng [12] mostraram que se existirem mais de um conjunto de potências de raízes consecutivas como um espaçamento específico, o peso estimado do código BCH pode ser melhorado. Este resultado será utilizado para demonstrar o Teorema 2.3.9.

Teorema 1.6.8. *Seja $g(x) \in \mathbb{F}_q[x]/ < x^n - 1 >$ o polinômio gerador de um código cíclico $\mathcal{C}$, de comprimento n, com $\mathbb{F}_q$ um corpo com q elementos. Se $g(\alpha^{l+i_1 a+i_2 c}) = 0$ para $i_1 = 0,1,2,\ldots,d_0-2$ e $i_2 = 0,1,\ldots,s$, com $\text{mdc}(n,a) = 1$ e $\text{mdc}(n,c) = 1$, então $d_H(\mathcal{C}) \geq d_0 + s$.*

1.7 Cotas Assintóticas

Um código $\mathcal{C}$ sobre um alfabeto $\mathcal{A}$ com q elementos ($q \geq 2$) possui três parâmetros fundamentais (n, M, d_H), que são o seu comprimento, seu número de palavras e sua distância mínima, respectivamente. Na Teoria dos Códigos, um problema estudado é a depência entre n, M e d_H, pois interessam os códigos que possuem M e d_H grandes relativamente a n.

Nessa seção vamos apresentar algumas limitantes dos tamanhos dos códigos demonstrados em [13] e [15].

Teorema 1.7.1 (Cota de Singleton)**.** *Seja $\mathcal{C}$ um código com parâmetros (n, M, d_H), definido sobre um alfabeto $\mathcal{A}$ com q elementos. Então*

$$M \leq q^{n-d_H+1}.$$

Definição 1.7.2. *Quando tivermos um código em que $d_H = n-k+1$, chamaremos esse código de* **MDS** *(Maximum Distance Separable).*

Definição 1.7.3. *Para todos os números naturais n, r e $q \geq 2$, define-se*

$$V_q(n, r) = \sum_{i=0}^{r} \binom{n}{i} (q-1)^i.$$

Observe que pelo Lema 1.1.6, $V_q(n, r) =\mid B(\mathbf{a}, n) \mid$, com $\mathbf{a} \in \mathcal{A}^n$.

Teorema 1.7.4 (Cota de Hamming)**.** *Se $\mathcal{C}$ é um código com parâmetros (n, M, d_H), e se $\kappa = \left\lfloor \frac{d_H-1}{2} \right\rfloor$, então*

$$M \cdot V_q(n, \kappa) \leq q^n,$$

com a igualdade valendo se, e somente se, $\mathcal{C}$ é perfeito.

Considere a função:

$$A(n, d_H) = \max\{M; \text{existe um código com parâmetros } (n, M, d_H)\}.$$

Corolário 1.7.5. *Para todos os números naturais n e d_H, temos*

$$A(n, d_H) \leq \frac{q^n}{V_q(n, \kappa)}.$$

Note que a cota de Singleton (Teorema 1.7.1) fornece a seguinte desigualdade

$$A(n, d_H) \leq q^{n-d_H+1}.$$

Definição 1.7.6. *Um código $\mathcal{C}$ de comprimento n e distância mínima d_H, tal que $\mid C \mid = A(n, d_H)$, será chamado de* **código ótimo**.

No próximo resultado apresentamos um limitante inferior para $A(n, d_H)$, que será utilizada para produzir uma cota assintótica.

Teorema 1.7.7 (Cota de Gilbert-Varshamov)**.** *Existe um código q-ário $\mathcal{C}$ de comprimento n, distância mínima d_H com*

$$A(n, d_H) \geq \frac{q^n}{V_q(n, d_H - 1)}.$$

As demonstrações dos Teoremas 1.7.4 e 1.7.7 foram omitidas, pois as demonstrações das Proposições 2.5.6 e 2.5.9 são análogas.

Cotas Assintóticas

Nessa seção vamos estudar a cota de Gilbert-Varshamov quando o comprimento do código tende para o infinito. O resultado é chamado de *cota assintótica*.

Definição 1.7.8. *Dado um código $\mathcal{C}$ com parâmetros (n, M, d_H) sobre um alfabeto $\mathcal{A}$ com q elementos. Definimos a* **distância relativa** *como sendo o número*

$$\delta = \frac{d_H}{n}$$

e a **taxa assintótica** *como sendo*

$$R(\delta) = \lim_{n\to\infty} \sup \frac{\log_q A(n, \delta n)}{n}.$$

A distância relativa é a medida da capacidade de correção de erros de um código. O valor exato de $R(\delta)$ não é conhecido. Vamos apresentar um limite superior para esta função, mostrando que todas as famílias com distância relativa aproximando de δ tem taxa assintótica aproximando desse limite.

Definição 1.7.9. *Define-se a* **função entropia** H_q *no intervalo* $\left[0, \frac{q-1}{q}\right]$ *como sendo*

$$H_q = \begin{cases} 0, & se \quad x = 0 \\ \log_q \frac{(q-1)^x}{x^x(1-x)^{1-x}}, & se \quad 0 < x \leq \frac{q-1}{q}. \end{cases}$$

Lema 1.7.10. *Para $0 \leq \delta \leq (q-1)/q$ e, para qualquer inteiro n, temos*

(a) $\log_q(V_q(n, \delta n)) \leq nH_q(\delta)$;

(b) $\lim_{n\to\infty} \frac{\log_q(V_q(n, \lfloor \delta n \rfloor))}{n} = H_q(\delta)$.

Teorema 1.7.11 (Cota Assintótica de Gilbert-Varshamov)**.** *Para $\delta \in \left[0, \frac{q-1}{q}\right]$, temos*

$$R(\delta) \geq 1 - H_q(\delta).$$

1.8 Transformada Discreta de Fourier

Nesta seção definimos a transformada discreta de Fourier e fazemos duas observações de acordo com [1, Cap. 1], e [2].

Definição 1.8.1. *Sejam $\mathbb{F}_q$ um corpo com q elementos e K o corpo de fatoração de x^n-1 sobre $\mathbb{F}_q$, isto é, o menor corpo no qual x^n-1 se fatora em fatores lineares. Então K contém uma raiz n-ésima primitiva da unidade α tal que $x^n-1=\prod_{i=0}^{n-1}(x-\alpha^i)$. Para um polinômio $f(x)\in\mathbb{F}_q[x]/<x^n-1>$, definimos a* **Transformada Discreta de Fourier (DFT)** *do polinômio f com respeito a α por*

$$\varphi_{\alpha,f}(x)=\sum_{j=0}^{n-1}f(\alpha^j)x^j.$$

A transformada discreta de Fourier $\varphi_\alpha : \mathbb{F}_q[x]\longrightarrow\mathbb{F}_q[x]$ é um isomorfismo de anéis e a **transformada inversa de Fourier** é definida por

$$\varphi_{\alpha,g}^{-1}(x)=\frac{1}{n}\sum_{i=0}^{n-1}g(\alpha^{-i})x^i.$$

Observe que a DFT pode ser vista como um polinômio ou como uma sequência $(\hat{C}_0,\hat{C}_1,\ldots,\hat{C}_{n-1})$, com $\hat{C}_j=f(\alpha^j)$ o j^{th} coeficiente da DFT.

Observação 1.8.2. *O polinômio $f(x)=a_0+a_1x+\ldots+a_{n-1}x^{n-1}$ tem um zero em α^j se, e somente se, $\hat{C}_j=0$. O polinômio $\varphi_{\alpha,f}(x)=\sum_{j=0}^{n-1}f(\alpha^j)x^j$ tem um zero em α^{-i} se, e somente se, $a_i=0$.*

Observação 1.8.3. *Como a Transformada da Discreta de Fourier é um isomoforfismo que leva $f(x)$ em $\varphi_{\alpha,f}(x)$, temos $\varphi^{-1}_{\alpha,\varphi_{\alpha,f}}(x)=f(x)$.*

1.9 Códigos Concatenados

Nesta seção definimos os *códigos concatenados* e apresentamos um algoritmo de decodificação para esta classe de códigos, como feito em [23].

Definição 1.9.1. *Sejam $\mathcal{A}$ um alfabeto finito, Φ um conjunto finito. Considere $\mathcal{C}_{in}$ um $(n,\mid\Phi\mid,d_H)$-código interno sobre $\mathcal{A}$, $\mathcal{C}_{out}$ um (N,M,D_H)-código externo sobre Φ e $\varepsilon_{in}:\Phi\rightarrow\mathcal{C}_{in}$ uma aplicação bijetora. O* **código contatenado** *$\mathcal{C}_{cont}=(\varepsilon_{in},\mathcal{C}_{out})$ consiste de todas as palavras em $\mathcal{A}^{nN}$ da forma*

$$(\varepsilon_{in}(z_1)\mid\varepsilon_{in}(z_2)\mid\ldots\mid\varepsilon_{in}(z_N)),$$

com $(z_1\ z_2\ \ldots\ z_N)$ *representando todas as palavras do código* $\mathcal{C}_{out}$.

O código $\mathcal{C}_{cont}$ é um $(nN, M, \geq d_H \cdot D_H)$-código sobre $\mathcal{A}$.

1.9.1 Decodificação de códigos concatenados

Seja $\mathcal{C}_{cont}$ um $(nN, M, \geq d_H \cdot D_H)$-código concatenado sobre $\mathcal{A}$ construído usando um (N, M, D_H)-código externo $\mathcal{C}_{out}$ sobre Φ e uma bijeção $\varepsilon_{in} : \Phi \to \mathcal{C}_{in}$ para um $(n, \mid \Phi \mid, d_H)$-código interno $\mathcal{C}_{in}$ sobre $\mathcal{A}$. Apresentamos agora um algoritmo de decodificação que corrige até $d_H \cdot D_H/2$ erros.

Suponha que uma palavra

$$y = (y_1 \mid y_2 \mid \ldots \mid y_N) \in \mathcal{A}^{nN}$$

tenha sido recebida, com cada bloco y_j em $\mathcal{A}^n$ e $d_H(c, y) \leq d_H \cdot D_H/2$.

Passo 1. Para cada $j = 1, 2, \ldots, N$, seja $\hat{c}_j$ a palavra mais próxima de y_j em $\mathcal{C}_{in}$ e seja $\hat{z}_j$ o valor de $\varepsilon_{in}^{-1}(\hat{c}_j)$.

Passo 2. Defina a palavra

$$x = (x_1\ x_2\ \ldots\ x_N) \in (\Phi \cup \{?\})^N,$$

com

$$x_j = \begin{cases} \hat{z}_j, & \text{se } d_H(y_j, \hat{c}_j) < \lfloor d_H/2 \rfloor \\ ?, & \text{caso contrário} \end{cases}.$$

Passo 3. Utilize um decodificador de erros e exclusões de $\mathcal{C}_{out}$ em x, produzindo uma palavra $(z_1, z_2, \ldots, z_N) \in \mathcal{C}_{out}$ ou um indicador de erro.

Passo 4. Se o decodificador do Passo 3 funcionar, tome

$$c = (\varepsilon_{in}(z_1) \mid \varepsilon_{in}(z_2) \mid \ldots \mid \varepsilon_{in}(z_N)).$$

Passo 5:

(i) Se $d_H(y, c) < d_H \cdot D_H/2$, então c foi a palavra enviada.

(ii) Se nenhuma palavra foi produzida no Passo 3, então mais de $d_H \cdot D_H/2$ foram cometidos e o decodificador indicará que houve falha na decodificação.

1.10 Função Piso e Teto

Nesta seção vamos revisar a definição e as principais propriedades da *função piso* (*menor inteiro*) e da *função teto* (*maior inteiro*) como feito em [22]. Estas funções são bastante utilizadas no Capítulo 3

Definição 1.10.1. *A* **função piso** *ou* **menor inteiro** *de um número real x é o maior inteiro menor ou igual a x. A função piso é denotada por $\lfloor x \rfloor$. E a* **função teto** *ou* **maior inteiro** *de um número real x é o menor inteiro maior ou igual a x. O valor da função teto é denotada por $\lceil x \rceil$.*

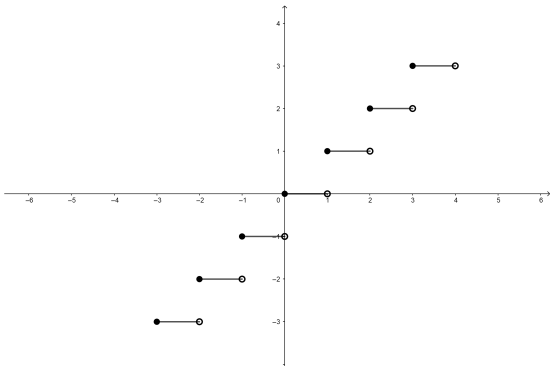


Figura 1.1: Função Piso $\lfloor x \rfloor$

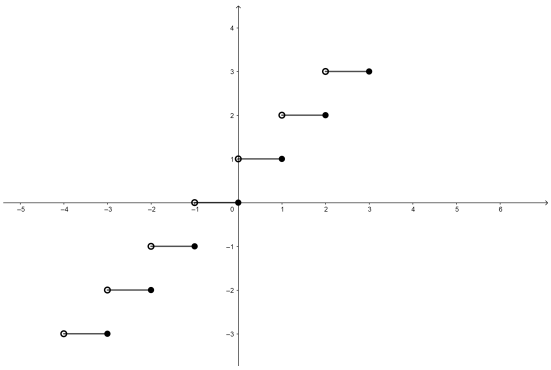


Figura 1.2: Função Teto $\lceil x \rceil$

Propriedades: Sejam $n \in \mathbb{Z}$ e $x \in \mathbb{R}$.

1. $\lfloor x \rfloor = x$ se, e somente se, $x \in \mathbb{Z}$.
2. $\lceil x \rceil = x$ se, e somente se, $x \in \mathbb{Z}$.
3. $\lfloor x \rfloor = n$ se, e somente se, $n \leq x < n + 1$.
4. $\lceil x \rceil = n$ se, e somente se, $n - 1 < x \leq n$.
5. $\lfloor x \rfloor = n$ se, e somente se,$x - 1 < n \leq x$.
6. $\lceil x \rceil = n$ se, e somente se, $x \leq n < x + 1$.
7. Para todos $x, y \in \mathbb{R}$, se $x < y$ então $\lfloor x \rfloor \leq \lfloor y \rfloor$.
8. Para todos $x, y \in \mathbb{R}$, se $x < y$ então $\lceil x \rceil \leq \lceil y \rceil$.
9. $x - 1 < \lfloor x \rfloor \leq x \leq \lceil x \rceil < x + 1$.
10. $\lfloor -x \rfloor = -\lceil x \rceil$.
11. $\lceil -x \rceil = -\lfloor x \rfloor$.

12. $\lfloor x + n \rfloor = \lfloor x \rfloor + n$.

13. $\lceil x + n \rceil = \lceil x \rceil + n$.

14. Se a e n são dois inteiros positivos, então $n \cdot \left\lfloor \frac{a}{n} \right\rfloor \leq a$.

Capítulo 2

Códigos para Canais de Leitura de Pares de Símbolos

Neste capítulo descrevemos a teoria básica dos *códigos para canais de leitura de pares de símbolos* de acordo com [3], [4] [5], [8], [16] e [28]. Para isto, usamos como alfabeto, um conjunto $\mathcal{A}$ finito com q elementos. Iniciamos definindo vetor de pares de símbolos e códigos de pares de símbolos. Determinamos a distância de pares, o peso de pares, a distância mínima e o peso mínimo de um código de pares de símbolos. Relacionamos a distância de Hamming de um código clássico com a distância de pares. Provamos uma cota de Singleton para códigos de pares, definimos os códigos de pares MDS (*Maximum Distance Separable*) e mostramos uma relação entre códigos MDS clássicos, com distância de Hamming estritamente menor que o comprimento do código, e os códigos de pares MDS. Mostramos a capacidade de correção de códigos de pares de símbolos. Fazemos algumas contruções de códigos de pares a partir da métrica de Hamming usando códigos intercalados e códigos cíclicos. Na última seção, mostramos alguns limitantes do tamanho de códigos de pares de símbolos.

2.1 Códigos de Pares de Símbolos

Definição 2.1.1. *Seja* $x = (x_0, ..., x_{n-1})$ *um vetor em* $\mathcal{A}^n$. *O* **vetor de pares de símbolos** *de* x *é definido por*

$$\pi(x) = [(x_0, x_1), (x_1, x_2), ..., (x_{n-2}, x_{n-1}), (x_{n-1}, x_0)].$$

Para todos $x, y \in \mathcal{A}$, temos

$$\pi(x + y) = \pi(x) + \pi(y).$$

Todo vetor $x \in \mathcal{A}^n$ tem uma representação $\pi(x) \in (\mathcal{A}, \mathcal{A})^n$. Entretanto, nem todo vetor u de pares em $(\mathcal{A}, \mathcal{A})^n$ tem um vetor correspondente em $\mathcal{A}^n$, porque u pode ter símbolos diferentes em posições de dois pares consecutivos. Por exemplo, o vetor de pares

$$u = [(1, \underset{\nwarrow}{1}) \quad , \quad (\underset{\nearrow}{0}, 2) \quad , (2,1), (1,0), (0,1)] \in (\mathbb{F}_3, \mathbb{F}_3)^5$$
$$x_1$$

não tem um vetor correspondente em $\mathbb{F}_3^5$, pois $x_1 = 1$ na direita do primeiro par, mas $x_1 = 0$ na esquerda do segundo par.

Definição 2.1.2. *Um vetor de pares de símbolos $y \in (\mathcal{A}, \mathcal{A})^n$ para o qual existe um vetor $x \in A^n$ tal que $\pi(x) = y$ é dito* **consistente**.

Definição 2.1.3. *O* **código de pares de símbolos** *de um código $\mathcal{C}$ é o código*

$$\pi(\mathcal{C}) = \{\pi(c); c \in \mathcal{C}\}.$$

Estamos interessados nos casos em que alguns pares lidos sejam versões corrompidas de pares de símbolos verdadeiros. O principal modelo de erro considerado para vetores de pares de símbolos é quando o número de pares errados é limitado por um inteiro t, definido como *t-pares de erro* a seguir.

Definição 2.1.4. *Seja $x = (x_0, \cdots, x_{n-1})$ um vetor em $\mathcal{A}^n$. Um vetor de pares $u = [(u_{0,0}, u_{0,1}), (u_{1,0}, u_{1,1}) \cdots, (u_{n-1,0}, u_{n-1,1})]$ é o* **resultado de t-pares de erros a partir de** *x se $\mid \{i : (u_i, u_{i+1}) \neq (x_i, x_{i+1})\} \mid \leq t$. Os índices são tomados módulo n e $(a, b) = (c, d)$ se ambos $a = c$ e $b = d$.*

Após definir o modelo de erro de pares de símbolos, o próximo passo naturalmente é provar condições necessárias e suficientes no código para alcançar corretabilidade dos erros de pares de símbolos. Para isto precisamos da definição de distância de pares que descrevemos a seguir.

Definição 2.1.5. *Sejam $u, v \in \mathcal{A}^n$. A* **distância de pares** *entre u e v é definida por*

$$d_P(u, v) = d_H(\pi(u), \pi(v)) = \mid \{i; (x_i, x_{i+1}) \neq (y_i, y_{i+1}) \text{ e } i = 0, \ldots, n-1\} \mid,$$

com $d_H(.,.)$ denotando a distância de Hamming entre dois vetores.

Exemplo 2.1.6. *A distância de pares entre os vetores* (12010) *e* (10112) *sobre* $\mathcal{A} = \{0, 1, 2\}$ *é dada por*

$$d_P(10102, 10112) = d_H(\pi(10102), \pi(10112)) =$$

$$d_H([(1,0),(0,1),(1,0),(0,2),(2,1)],[(1,0),(0,1),(1,1),(1,2),(2,1)]) = 2.$$

Proposição 2.1.7. *Dados* $x, y, w \in \mathcal{A}^n$*, valem as seguintes propriedades:*

i) Positividade: $d_P(x, y) \geq 0$*, com* $d_P(x, y) = 0$ *se, e somente se,* $x = y$*.*

ii) Simetria: $d_P(x, y) = d_P(y, x)$*.*

iii) Desigualdade Triangular: $d_P(x, y) \leq d_P(x, w) + d_P(w, y)$*.*

Demonstração. Os dois primeiros itens são óbvios. Para provar a desigualdade triangular, observe que se $(x_i, x_{i+1}) \neq (y_i, y_{i+1})$, para algum $i = 0, ..., n-1$ e $n = 0$ (módulo n), então no mínimo $(x_i, x_{i+1}) \neq (w_i, w_{i+1})$ ou $(w_i, w_{i+1}) \neq (y_i, y_{i+1})$ deve ser satisfeito, caso contrário teríamos $(x_i, x_{i+1}) = (w_i, w_{i+1})$ e $(w_i, w_{i+1}) = (y_i, y_{i+1})$, o que implicaria $(x_i, x_{i+1}) = (y_i, y_{i+1})$, contradizendo a hipótese. Logo, $d_P(x, y) \leq d_P(x, w) + d_P(w, y)$. □

Assim, a distância de pares é uma métrica em $(\mathcal{A}, \mathcal{A})^n$.

Definição 2.1.8. *Seja* $\mathcal{C} \subset \mathcal{A}^n$ *um código. A* **distância mínima de pares** *de* $\mathcal{C}$ *é o inteiro*

$$d_P = d_P(\mathcal{C}) = min\{d_P(u, v);\ u, v \in C \ e\ u \neq v\}.$$

Exemplo 2.1.9. *Seja* $\mathcal{A} = \{0, 1\}$*. Considere o código binário*

$$\mathcal{C} = \{0000, 1010, 1011, 1001\} \subset \mathcal{A}^4.$$

Vamos calcular as distâncias de pares entre os elementos de $\mathcal{C}$*.*

$$d_P(1010, 1011) = d_H(\pi(1010), \pi(1011)) =$$

$$d_H([(1,0),(0,1),(1,0),(0,1)],[(1,0),(0,1),(1,1),(1,1)]) = 2$$

$$d_P(0000, 1011) = 4, \quad d_P(0000, 1001) = 3, \quad d_P(0000, 1010) = 4,$$

$$d_P(1010, 1001) = 3, \quad d_P(1011, 1001) = 2.$$

Logo a distância mínima de pares de $\mathcal{C}$ *é* $d_P = 2$*.*

A relação entre a distância de pares e a distância de Hamming é dada a seguir.

Proposição 2.1.10. *Para $x, y \in \mathcal{A}^n$, seja $0 < d_H(x,y) < n$ a distância de Hamming. Então*

$$d_H(x,y) + 1 \leq d_P(x,y) \leq 2d_H(x,y).$$

Para $d_H(x,y) = 0$ ou n, tem-se $d_P(x,y) = d_H(x,y)$.

Demonstração. Defina os conjuntos $S_H = \{j;\ x_j \neq y_j\}$ e $S_P = \{i;\ (x_i, x_{i+1}) \neq (y_i, y_{i+1})\}$, então $\mid S_H \mid = d_H(x,y)$ e $\mid S_P \mid = d_P(x,y)$. Cada índice em S_H aparece em exatamente dois pares de S_P, isto é, para cada contribuição 1 à $d_H(x,y)$ temos uma contribuição de 2 à $d_P(x,y)$. Logo,

$$d_P(x,y) \leq 2d_H(x,y).$$

Note que se $d_H(x,y) < n$, existe no mínimo um par com apenas um dos índices $i,\ i+1$ em S_H, isto é, se $d_H(x,y) < n$, existe $0 \leq i \leq n-1$ tal que $i \notin S_H$ e $i+1 \in S_H$, então $(x_i, x_{l+i}) \neq (y_i, y_{l+i})$ e $i \in S_P$. Logo, existe uma coordenada que contribui com uma unidade à $d_P(x,y)$ e não contribui com $d_H(x,y)$, portanto

$$d_H(x,y) + 1 \leq d_P(x,y).$$

□

Corolário 2.1.11. *Se $\mathcal{C}$ é um código tal que $0 < d_H(\mathcal{C}) < n$, então*

$$d_H(\mathcal{C}) + 1 \leq d_P(\mathcal{C}) \leq 2d_H(\mathcal{C}).$$

Demonstração. Por definição, $d_H(\mathcal{C}) \leq d_H(x,y)$, para todo $x, y \in \mathcal{C}$ com $x \neq y$. Assim, pela Proposição 2.1.10, $d_H(\mathcal{C}) + 1 \leq d_H(x,y) + 1 \leq d_P(x,y)$, para todo $x, y \in \mathcal{C}$ com $x \neq y$, em particular, $d_H(\mathcal{C}) + 1 \leq d_P(\mathcal{C})$. Além disso, $d_P(\mathcal{C}) \leq d_P(x,y) \leq 2d_H(x,y)$, para todo $x, y \in \mathcal{C}$, em particular, $d_P(\mathcal{C}) \leq 2d_H(\mathcal{C})$. □

Definição 2.1.12. *Seja $u \in \mathcal{A}^n$. O* **peso de pares** *de u é definido por*

$$\omega_P(u) = \omega_H(\pi(u)) = \mid \{i; (x_i, x_{i+1}) \neq (0,0)\ e\ i = 0, \ldots, n-1\} \mid .$$

Definição 2.1.13. *Seja $\mathcal{C} \subset \mathcal{A}^n$ um código. O* **peso mínimo de pares** *de $\mathcal{C}$ é o número*

$$\omega_P(\mathcal{C}) = min\{\omega_P(u); u \in C \setminus \{\mathbf{0}\}\}.$$

De forma análoga a Proposição 1.2.4 para códigos lineares clássicos, obtemos o seguinte resultado.

Proposição 2.1.14. *Seja $\mathcal{C} \subset \mathcal{A}^n$ um código linear, temos*

(i) $d_P(x,y) = \omega_P(x-y)$, *para todos* $x, y \in \mathcal{A}^n$.

(ii) $d_P(\mathcal{C}) = \omega_P(\mathcal{C})$.

Demonstração. O item *(i)* segue diretamente das definições de distância e peso de pares

$$\begin{aligned} d_P(x,y) &= d_H(\pi(x),\pi(y)) = \{i;\ (x_i, x_{i+1}) \neq (y_i, y_{i+1})\} \\ &= \{i;\ (x_i, x_{i+1}) - (y_i, y_{i+1}) \neq (0,0)\} \\ &= \{i;\ (x_i - y_i, x_{i+1} - y_{i+1}) \neq (0,0)\} \\ &= \omega_H(\pi(x-y)) = \omega_P(x-y). \end{aligned}$$

O item *(ii)* segue do fato que, para todos os vetores $x, y \in \mathcal{C}$ com $x \neq y$, tem-se $z = x - y \in \mathcal{C} \setminus \{\mathbf{0}\}$, pois $\mathcal{C}$ é um código linear. Assim,

$$\begin{aligned} d_P(\mathcal{C}) &= min\{d_P(x,y);\ x, y \in \mathcal{C} \text{ e } x \neq y\} = min\{\omega_P(x-y);\ x, y \in \mathcal{C} \text{ e } x \neq y\} \\ &= min\{\omega_P(z);\ z \in \mathcal{C} \setminus \{\mathbf{0}\}\} = \omega_P(\mathcal{C}). \end{aligned}$$

□

Observação 2.1.15. *Se $\mathcal{A} = \{0,1\}$, então $d_P(x,y) = \omega_P(x+y)$, pois $-y = y$ para todo $y \in \mathcal{A}$.*

Definição 2.1.16. *Sejam $x, y \in \mathcal{A}^n$. Defina $S_H = \{j; x_j \neq y_j\}$ e seja $S_H = \bigcup_{i=1}^{L} B_i$ a* **menor partição do conjunto S_H em subconjuntos de índices consecutivos**, *isto é, se s_l e e_l são tais que $1 < s_l < e_l < n$ e todos os símbolos estão entre s_l e e_l estão em S_H, então $B_l = [s_l, e_l] = \{s_l, s_{l+1}, s_{l+2}, ..., e_{l-1}, e_l\}$ e, se $n, 1 \in S_H$, vamos considerar os índices consecutivos antes de n e depois de 1 que também estão em S_H e, neste caso, teremos $B_l = [k_l, t_l] = \{k_l, k_{l+1}, ..., n-1, 0, 1, 2, ..., t_{l-1}, t_l\}$. Os índices são considerados módulo n.*

Exemplo 2.1.17. *Considere os seguintes casos:*

i) Sejam $x = (0,1,2,1,2,0,0,1,1,2,2)$ e $y = (1,2,2,0,1,0,1,1,1,2,1)$ em F_3^{11}. Temos $S_H = \{0,1,3,4,6,10\} = \bigcup_{i=1}^{3} B_i$, com $B_1 = [10,1] = \{0,1,10\}$, $B_2 = [3,4] = \{3,4\}$ e $B_3 = [6,6] = \{6\}$.

ii) Sejam $x = (0,1,0,0,1,1,1,1,0,1,0,0,0,1,0)$ e $y = (1,1,1,0,1,1,0,0,1,1,1,1,0,0,1)$ em F_2^{15}. Temos $S_H = \{0,2,6,7,8,10,11,13,14\} = \bigcup_{i=1}^{4} B_i$, com $B_1 = [13,0] = \{0,13,14\}$, $B_2 = [2,2] = \{3\}$, $B_3 = [6,8] = \{6,7,8\}$ e $B_4 = [10,11] = \{10,11\}$.

Teorema 2.1.18. *Para duas palavras $x, y \in \mathcal{A}^n$, com $0 < d_H(x,y) < n$, considere o conjunto $S_H = \{j; x_j \neq y_j\}$ e $S_H = \bigcup_{l=1}^{L} B_l$ a menor partição de S_H em subconjuntos de índices consecutivos, conforme descrito na Definição 2.1.16. Então*

$$d_P(x,y) = d_H(x,y) + L.$$

Demonstração. O fato da partição ser mínima garante que não existem dois índices consecutivos l, $l+1$ que pertencem a diferentes subconjuntos de S_H (se existisse tais subconjuntos poderiam ser unidos resultando em uma partição menor). Por esse motivo, a distância de pares entre x e y pode ser calculada como a soma dos tamanhos dos subconjuntos dos pares $\beta_l = \{(s_{l-1}, s_l), (s_l, s_{l+1}), (s_{l+1}, s_{l+2}), ..., (e_{l-1}, e_l), (e_l, e_{l+1})\}$. Isto acontece pois o par (s_{l-1}, s_l), por exemplo, se refere aos pares $(x_{s_{l-1}}, x_{s_l})$, $(y_{s_{l-1}}, y_{s_l})$ que são diferentes, já que $s_l \in S_H$, ou seja, o par (s_{l-1}, s_l) contribui com uma unidade a $d_P(x,y)$. E isso acontece com os demais elementos dos subconjuntos de pares β_l. Note ainda que o número de pares em cada subconjunto de pares β_l é igual a $|B_l| + 1$, portanto a soma nos fornece

$$d_P(x,y) = \sum_{l=1}^{L} |B_l| + L = d_H(x,y) + L.$$

□

Corolário 2.1.19. *Para toda palavra não nula $x \in \mathcal{A}$,*

$$\omega_P(x) = \omega_H(x) + L.$$

Demonstração. $\omega_P(x) = d_P(x, \mathbf{0}) = d_H(x, \mathbf{0}) + L = \omega_H(x) + L.$ □

No Capítulo 1, definimos a cota de Singleton, que limita o tamanho de um código com distância de Hamming e comprimento fixos, os códigos MDS (*Maximum Distance Separable*). De forma análoga, demonstramos agora a cota de Singleton para código de pares e definimos os *códigos de pares de símbolos MDS*, como feito em [5].

Teorema 2.1.20 (Cota de Singleton)**.** *Se $\mathcal{C}$ é um código de pares q-ário de comprimento n e distância mínima de pares d_P, com $q \geq 2$ e $2 \leq d_P \leq n$, então*

$$| \mathcal{C} | \leq q^{n-d_P+2}.$$

Demonstração. Seja $\mathcal{C}$ um (n, d_P)-código de pares q-ário, com $q \geq 2$ e $2 \leq d_P \leq n$. Delete as últimas $d_P - 2$ coordenadas de todas as palavras de $\mathcal{C}$. Observe que qualquer $d_P - 2$ coordenadas consecutivas contribuem a no máximo $d_P - 1$ para a distância de pares. Como $\mathcal{C}$ tem distância de pares d_P, os vetores resultantes de comprimento $n - d_P + 2$

continuam distintos após deletar as últimas $d_P - 2$ coordenadas de todas as palavras. O número máximo de vetores distintos de comprimento $n - d_P + 2$ sobre um alfabeto de tamanho q é q^{n-d_P+2}. Logo, $\mid \mathcal{C} \mid \leq q^{n-d_P+2}$. □

Definição 2.1.21. *Chamamos um (n, d_P)-código de pares q-ário de tamanho q^{n-d_P+2} de* **maximum distance separable (MDS)**.

No estudo dos códigos corretores de erros clássicos, consideramos um código como um subconjunto próprio de $\mathcal{A}^n$, com $\mathcal{A}$ o alfabeto do código. A razão disto é que se $\mathcal{C} = \mathcal{A}^n$, não há como detectar se uma palavra chega ao destinatário com erro, uma vez que $d_H(\mathcal{C})$ neste caso é 1 (Teorema 1.1.8). No entanto, para códigos de pares de símbolos, $d_P(\mathcal{C}) = 2$, se $\mathcal{C} = \mathcal{A}^n$, então podemos considerar este caso.

Exemplo 2.1.22. *O código $\mathcal{C} = \mathcal{A}^n$ é um $(n, 2)$-código de pares MDS, para todo $n \geq 2$ e $q \geq 2$. De fato, dada quaisquer duas palavras diferentes e não nulas $x, y \in \mathcal{C}$, temos $d_H(x, y) \geq 1$. Assim, pela Proposição 2.1.10,*

$$d_P(x, y) \geq d_H(x, y) + 1 \geq 1 + 1 = 2.$$

Além disso, seja $c \in \mathcal{C}$ com apenas uma coordenada não nula, isto é, $\omega_H(c) = 1$. Então pelo Teorema 2.1.18, $d_P(c, \mathbf{0}) = d_H(c, \mathbf{0}) + L = 1 + 1 = 2$. Logo $d_P = 2$. Claramente $\mid \mathcal{C} \mid = \mid \mathcal{A}^n \mid = q^n = q^{n-d_P+2}$, portanto, $\mathcal{C}$ é um código de pares MDS.

Proposição 2.1.23. *Um (n, d_H)-código MDS, com $d_H < n$, é um $(n, d_H + 1)$-código de pares MDS.*

Demonstração. Seja $\mathcal{C}$ um (n, d_H) código q-ário de tamanho q^{n-d_H+1}. Pela Proposição 2.1.11, $d_P \geq d_H + 1$, que implica $n - d_P + 2 \leq n - d_H - 1 + 2 = n - d_H + 1$. Logo, como $f(x) = q^x$ é uma função crescente, segue $\mid \mathcal{C} \mid = q^{n-d_H+1} \geq q^{n-d_P+2}$. Portanto, pelo Teorema 2.1.20, $\mathcal{C}$ tem exatamente q^{n-d_P+2} elementos. Além disso, $\mid \mathcal{C} \mid = q^{n-d_P+2} = q^{n-d_H+1}$, assim $d_P = d_H + 1$. Portanto, $\mathcal{C}$ é um $(n, d_H + 1)$-código de pares MDS. □

A construção de códigos de pares MDS é interessante, pois são os códigos com melhor capacidade de correção para comprimento e dimensão fixos. Em [5], Chee *et al.* construíram infinitas famílias de códigos de pares de símbolos MDS. Eles utilizaram códigos intercalados, teoria de grafos e configurações combinatoriais para construir tais códigos. Kai, Zhu e Li [16] construíram códigos de pares de símbolos MDS a partir de códigos cíclicos e constacíclicos. Em [8] são feitas novas construções de códigos de pares de símbolos MDS a partir de códigos lineares sobre o corpo $\mathbb{F}_q$. Zhang [30] também constrói alguns códigos de pares MDS com certos parâmetros fixos. Além disso, Li e Ge [18] construíram três novas classes de códigos de pares de símbolos MDS com distância mínima de pares cinco

e seis, além de encontrar uma condição necessária e suficiente que garante que uma classe de códigos cíclicos são códigos de pares de símbolos MDS. Finalmente, Kai *et al.* [17] construíram três novos códigos de pares MDS com distância mínima de pares seis e sete a partir de códigos constacíclicos com raízes repetidas.

A distância mínima de um código está relacionada com a sua capacidade de detecção e correção de erros (Teorema 1.1.8). Nesta seção, provamos os principais resultados a respeito da capacidade de correção de erros de um código de pares de símbolos, como Cassuto e Blaum [3].

Proposição 2.1.24. *Um código C pode corrigir até t pares de erros se, e somente se, $d_P(\mathcal{C}) \geq 2t+1$.*

Demonstração. Suponha que o código contenha duas palavras u e v com distância de pares máxima $2t$ entre elas. Seja w um vetor de pares que coincide com $\pi(u)$ em todos os lugares em que $\pi(u)$ coincida com $\pi(v)$. Além disso, deixe w coincidir com $\pi(u)$ nos t primeiros lugares em que $\pi(u)$ e $\pi(v)$ são diferentes e com $\pi(v)$ nos lugares restantes (se $d_P(u,v) < t$, tome $w = \pi(u)$). Assim, $d_H(\pi(u), w) \leq t$ e $d_H(\pi(v), w) \leq t$. Agora, suponha que w é recebida junto com a informação de que no máximo t pares de erros ocorreram. Então u e v podem ter sido transmitidos (ou até mesmo outra palavra do código). Assim, não existe uma maneira simples de decidir qual palavra foi transmitida, logo existe uma falha ao corrigir até t pares de erros.

Reciprocamente, suponha que $d_P(\mathcal{C}) \leq 2t+1$ e uma palavra w foi recebida junto com a informação que um erro de pares de peso máximo t ocorreu. Se existem duas palavras u e v com distância entre $\pi(u), \pi(v)$ e w de no máximo t, então pela desigualdade triangular $d_P(u,v) = d_H(\pi(u), \pi(v)) \leq d_H(\pi(u), w) + d_H(w, \pi(v)) \leq 2t$, contradizendo a hipótese. Logo existe uma única palavra u com distância no máximo t entre $\pi(u)$ e w e, portanto, podemos deduzir que u foi transmitido. □

Teorema 2.1.25. *Se um código $\mathcal{C}$ pode corrigir todos os t pares de erros e as entradas do decodificador são vetores de pares consistentes então $d_P(\mathcal{C}) \geq 2t$.*

Note que para demonstrar a Proposição 2.1.24 foi necessário construir um vetor de pares w, a partir de duas palavras do código u, v com $d_P(u,v) \leq 2t$, de forma que $d_H(\pi(u), w) \leq t$ e $d_H(\pi(v), w) \leq t$. Para demostrar o Teorema 2.1.25, será necessário contruir um vetor z, a partir de duas palavras do código x, y com $d_P(x,y) \leq 2t-1$, de forma que $d_P(x,z) \leq t$ e $d_P(y,z) \leq t$, ou seja, como uma das hipóteses do Teorema 2.1.25 diz que o vetor recebido pelo decodificador é um vetor de pares consistente, é necessário encontrar um vetor $z \in \mathcal{A}^n$ e não mais um vetor de pares $w \in (\mathcal{A}, \mathcal{A})^n$. Por este motivo, precisamos do seguinte Lema para a demonstração do Teorema 2.1.25.

Lema 2.1.26. *Se $d_P(x,y) \leq 2t-1$, então existe uma palavra $z \in \mathcal{A}^n$ tal que $d_P(x,z) \leq t$ e $d_P(z,y) \leq t$.*

Demonstração. Se $d_P(x,y) < t$, tome $z = x$. Daí $d_P(x,z) = 0 \leq t$ e $d_P(z,y) \leq t$.

Para $d_P(x,z) \geq t$, defina $S_H = \{j;\ x_j \neq y_j\}$ e seja $S_H = \sum_{l=1}^{L} B_l$ a partição minimal de S_H em subconjuntos de índices consecutivos, como na Definição 2.1.16. Para a seleção de z, vamos precisar construir o conjunto T_H a partir de S_H. Defina o contador k e o inicialize em $k = t$.

Se existe um subconjunto $B_l = [s_l, e_l]$, com $k-1$ ou menos elementos, acrescente-o à T_H, subtraia seu tamanho mais um de k e assuma o resultado como o novo valor do contador k. Repita o processo para o novo valor de k até que não seja possível encontrar um subconjunto com tamanho menor que k. Note que, para o primeiro valor do contador $k = t$, se não existir um subconjunto com tamanho menor que k, passa-se direto para o próximo passo.

Agora, se $k = 0$ ou 1, pare o processo. Se $k \geq 2$, pegue qualquer $B_l = [s_l, e_l]$ de $S_H \setminus T_H$ e adicione $[s_l, s_l + k - 2]$ (de tamanho $k-1$) à T_H e pare o processo. Tome

$$z_j = \begin{cases} y_j, & \text{se } j \in T_H \\ x_j, & \text{caso contrário.} \end{cases}$$

Denote o número de subconjuntos da partição minimal de T_H por L_1 e o número de subconjuntos da partição minimal de $S_H \setminus T_H$ por L_2. Cada subconjunto adicionado a T_H contribui para o aumento da distância de pares entre x e z seu tamanho mais um. Logo, como o contador começa em t, temos:

i) Se o contador terminar em 1, $d_P(x,z) = t-1$ e nenhum dos subconjuntos minimais de S_H se dividem entre T_H e $S_H \setminus T_H$, pois foram incluídos apenas subconjuntos inteiros a T_H, assim $L_1 + L_2 = L$. Do Teorema 2.1.18, como $d_H(x,y) = \mid S_H \mid$, $d_H(x,z) = \mid T_H \mid$ e $d_H(y,z) = \mid S_H \mid - \mid T_H \mid$, temos

$$2t-1 \geq d_P(x,y) = \mid S_H \mid + L \qquad \text{e} \qquad t-1 = d_P(x,z) = \mid T_H \mid + L_1.$$

Assim,

$$\begin{aligned} d_P(y,z) &= \mid S_H \mid - \mid T_H \mid + L_2 \leq (2t-1-L) - (t-1-L_1) + L_2 \\ &= t - L + L_1 + L_2 = t - L + L = t. \end{aligned}$$

Logo, $d_P(x,z) = t-1 \leq t$ e $d_P(y,z) \leq t$.

ii) Para todas as demais situações, se o contador terminar em zero ou $k \geq 2$, temos pelo Teorema 2.1.18 $d_P(x,z) = t$. Como no máximo um dos L subconjuntos minimais de S_H se dividem entre T_H e $S_H \setminus T_H$, temos $L_1 + L_2 \leq L + 1$. Assim,

$$\begin{aligned} d_P(y,z) &= |\, S_H \,| - |\, T_H \,| + L_2 \leq (2t - 1 - L) - (t - L_1) + L_2 \\ &= t - 1 - L + L_1 + L_2 \leq t - 1 - L + L + 1 = t. \end{aligned}$$

Portanto, $d_P(x,z) = t$ e $d_P(y,z) \leq t$. □

Exemplo 2.1.27. *Sejam* $x = (02212110100210211102)$, $y = (10212121100210211101)$ *vetores de* $\{0,1,2\}^{20}$. *Então* $d_P(x,y) = 7 = 2t - 1$, *com* $t = 4$. *Vamos construir o vetor* $z \in \{0,1,2\}^{20}$ *tal que* $d_P(x,z) \leq t$ *e* $d_P(y,z) \leq t$ *usando a demonstração do Lema 2.1.26. Para isto, seja* $S_H = \{0,1,6,7,19\} = [19,1] \cup [6,7]$ *o conjunto dos índices em que* x *e* y *se diferem. Iniciamos o contador* $k = t = 4$, *podemos escolher qualquer um dos intervalos* $[19,1]$ *ou* $[6,7]$, *pois ambos tem tamanho menor ou igual a* $k - 1 = 3$. *Vamos analisar as duas escolhas:*

1. *Coloque* $[19,1] \subset T_H$. *Subtraindo seu tamanho mais um, obtemos o novo valor de* $k = 0$. *Então paramos o processo e fazemos* $z = (10212110100210211101)$. *Então* $d_P(x,z) = 4 = t$ *e* $d_P(y,z) = 3 \leq t$, *exatamente como queríamos.*

2. *Coloque* $[6,7] \subset T_H$. *Subtraindo seu tamanho mais um, obtemos o novo valor de* $k = 1$. *Então paramos o processo e fazemos* $z = (02212121100210211102)$. *Então* $d_P(x,z) = 3 \leq t$ *e* $d_P(y,z) = 4 = t$.

Exemplo 2.1.28. *Sejam* $x = (011101000011011110)$ *e* $y = (010010111011011110)$ *vetores de* $\{0,1\}^{18}$. *Então* $S_H = \{2,3,4,5,6,7,8\} = [2,8]$ *e* $d_P(x,y) = 8 \leq 2t - 1$, *com* $t = 5$. *Iniciando o contador em* $k = t = 5$, *note que não existe nenhum subcojunto de* S_H *com tamanho menor ou igual a* $k-1 = 4$, *então fazemos* $T_H = [2,5]$ *e* $z = (010010000011011110)$. *Assim,* $d_P(x,z) = 5 = t$ *e* $d_P(y,z) = 4 \leq t$.

O Lema 2.1.26 aqui apresentado foi uma alteração do [3, Lema 5] no qual alteramos as hipóteses para o que precisamos na demonstração do Teorema 2.1.25. O [3, Lema 5] diz que se $d_P(x,y) = 2t - 1$, então existe $z \in \mathcal{A}^n$ tal que $d_P(x,z) = t$ e $d_P(z,y) \leq t$. Fizemos esta alteração, pois o algoritmo proposto na demonstração do [3, Lema 5] não chegava ao resultado exato para todos os casos. O exemplo que apresentamos agora justifica esta nossa consideração.

Para $x = (02212110100210211102)$ e $y = (10212121100210211101)$ vetores de $\{0,1,2\}^{20}$, temos $d_P(x,y) = 7 = 2 \cdot 4 - 1$, $S_H = \{0,1,6,7,19\} = [19,1] \cup [6,7]$. Pela demonstração proposta por [3], para construir a palavra z precisamos construir um conjunto T_H a partir de S_H. Para isto, inicializamos o contador em $k = 4$ e tomamos um subconjunto com tamanho $k - 1 = 3$ ou menor e adicionamos à T_H. Neste exemplo, podemos utilizar

qualquer um dos subconjuntos de índices consecutivos. Adicionamos $[6, 7]$ à T_H. Daí, temos um novo valor do contador $k' = 4 - 3 = 1$. Como não existe um subconjunto com tamanho menor que k', passamos para o último passo do algoritmo que seria adicionar um subconjunto de tamanho $k' - 1$ a partir de outro subconjunto qualquer, mas $k' - 1 = 0$. Assim, $T_H = \{6, 7\}$ e daí, pelo algoritmo, temos $z = (02212121100210211102)$, com $d_P(x, z) = 3 < t = 4$ e $d_P(y, z) = 4$.

Se, no último passo do algoritmo, adicionamos o subconjunto $[19, 19+1-2] = [18, 19]$ à T_H, teríamos $T_H = \{6, 7, 18, 19\}$ e $z = (02212121100210211101)$. Daí, $d_P(x, z) = 5 > t$ e $d_P(y, z) = 3$, o que também não funcionaria.

Observamos que a maneira como Cassuto e Blaum, em [3], descrevem a finalização do algoritmo, não deixa claro qual deve ser o procedimento exato neste passo do algoritmo. No exemplo acima, as duas possíveis interpretações levam a valores diferentes da distância de pares entre x e z que não atingem os resultados descritos no Lema.

A prova do Teorema 2.1.25 segue do Lema 2.1.26.

Demonstração. (**Demonstração do Teorema 2.1.25**) Suponha que o código contenha duas palavras x e y com distância de pares máxima $2t - 1$ entre elas. Pelo Lema 2.1.26, existe $z \in \mathcal{A}^n$ tal que $d_P(x, z) \leq t$ e $d_P(y, z) \leq t$. Suponha que um vetor de pares consistentes $\pi(z)$ seja recebida junto com a mensagem de que no máximo t pares de erros ocorreram. Então x e y podem ter sido transmitidos (ou até mesmo outra palavra do código). Assim, não existe uma maneira simples de decidir qual palavra foi transmitida, logo existe uma falha ao corrigir até t-pares de erros. □

2.2 Construções a partir da métrica de Hamming

Na Teoria de Códigos, considerar símbolos adjacentes como pares provém do problema de correção de erros em grande quantidade (*error bursts*). Assim, não nos surpreende que o procedimento de intercalar, clássico nos métodos de correção de *error bursts*, seja útil para o problema em códigos de pares de símbolos.

Na Proposição 2.1.10 provamos que a distância de pares excede a distância de Hamming em pelo menos uma unidade. Como consequência, a correção dos pares de erros não é tão eficiente nos códigos de pares construídos a partir de códigos lineares. Para melhorar o processo de correção, utilizam-se códigos intercalados que definimos nesta seção que possuem $d_P(\mathcal{C}) = 2d_H(\mathcal{C})$, como demonstramos no Teorema 2.2.3.

Definição 2.2.1. *Sejam $\mathcal{C}_1$ um (n, M_1, d_H)-código e $\mathcal{C}_2$ um (n, M_2, d_H)-código. Um* **código intercalado** *é obtido mesclando as palavras dos códigos $\mathcal{C}_1$ e $\mathcal{C}_2$ da seguinte*

maneira, para $(x_0, x_1, \ldots, x_{n-1}) \in \mathcal{C}_1$ *e* $(y_0, y_1, \ldots, y_{n-1}) \in \mathcal{C}_2$ *uma palavra do código intercalado é*

$$(x_0, y_0, x_1, y_1, \ldots, x_{n-1}, y_{n-1}).$$

Exemplo 2.2.2. *Sejam* $\mathcal{C}_1 = \{(0,0,0,0)\}$ *e* $\mathcal{C}_2 = \{(1,1,1,1),(2,2,2,2)\}$. *Então o código intercalado obtido mesclando* $\mathcal{C}_1$ *e* $\mathcal{C}_2$ *será*

$$\{(0,1,0,1,0,1,0,1),(0,2,0,2,0,2,0,2)\}.$$

Teorema 2.2.3. *Um código intercalado* $\mathcal{C}$ *obtido mesclando um* (n, M_1, d_H)*-código* $\mathcal{C}_1$ *e um* (n, M_2, d_H)*-código* $\mathcal{C}_2$ *é um* $(2n, M_1M_2, d_H)$*-código com* $d_P = 2d_H$.

Demonstração. É claro que o código $\mathcal{C}$ tem comprimento $2n$ e M_1M_2 palavras.

Sejam $u = (x_0, y_0, \ldots, x_{n-1}, y_{n-1})$ e $v = (a_0, b_0, \ldots, a_{n-1}, b_{n-1})$ palavras de $\mathcal{C}$ com $(x_0, \ldots, x_{n-1}), (a_0, \ldots, a_{n-1}) \in \mathcal{C}_1$ e $(y_0, \ldots, y_{n-1}), (b_0, \ldots, b_{n-1}) \in \mathcal{C}_2$. Então

$$\begin{aligned} d_H(u,v) &= |\,\{i;\ u_i \neq v_i \text{ e } i = 1, \ldots, 2n\}\,| \\ &= |\,\{i;\ x_i \neq a_i \text{ e } i = 1, \ldots, n\}\,| + |\,\{j;\ y_j \neq b_i \text{ e } j = 1, \ldots, n\}\,|, \end{aligned}$$

ou seja,

$$d_H(u,v) = d_H(x,a) + d_H(y,b), \tag{2.1}$$

logo a distância mínima de Hamming de $\mathcal{C}$ é

$$\begin{aligned} d_H(\mathcal{C}) &= \min\{d_H(u,v);\ u,v \in \mathcal{C} \text{ e } u \neq v\} \\ &= \min\{d_H(x,a) + d_H(y,b);\ x,y \in \mathcal{C}_1,\ a,b \in \mathcal{C}_2 \text{ e, } x \neq a \text{ ou } y \neq b\} \\ &= \min\{d_H(\mathcal{C}_1), d_H(\mathcal{C}_2)\} = d_H. \end{aligned}$$

Vamos calcular $d_P(\mathcal{C})$. Para isto, dadas duas palavras de $\mathcal{C}$ não nulas $u = (x_0, y_0, \ldots, x_{n-1}, y_{n-1})$ e $v = (a_0, b_0, \ldots, a_{n-1}, b_{n-1})$ com $(x_0, \ldots, x_{n-1})$, $(a_0, \ldots, a_{n-1}) \in \mathcal{C}_1$ e $(y_0, \ldots, y_{n-1})$, $(b_0, \ldots, b_{n-1}) \in \mathcal{C}_2$, considere dois casos.

i) Pelo menos um dos conjuntos de índices consecutivos descritos na Definição 2.1.16 tem 2 ou mais elementos. Isso significa que existe um índice $j = 0, \ldots, n-1$ tal que ou $x_j \neq a_j$ e $y_j \neq b_j$, ou $y_j \neq b_j$ e $x_{j+1} \neq a_{j+1}$, para ambos os casos, temos $x \neq a$ e $y \neq b$, logo $d_H(x,a) \geq d_H$ e $d_H(y,b) \geq d_H$. Assim, por (2.1), $d_H(u,v) = d_H(x,a) + d_H(y,b) \geq 2d_H$. Então, pelo Teorema 2.1.18,

$$d_P(u,v) \geq d_H(u,v) + 1 \geq 2d_H + 1.$$

ii) Todos os subconjuntos de índices consecutivos descritos na Definição 2.1.16 possuem apenas um elemento. Assim, como $d_H(\mathcal{C}) = d_H$, temos $L \geq d_H$, logo pelo Teorema 2.1.18

$$d_P(u, v) = d_H(u, v) + L \geq d_H + d_H = 2d_H.$$

Portanto, para quaisquer $u, v \in \mathcal{C}$ tais que $0 < d_H(u, v) < n$, temos $d_P(u, v) \geq 2d_H$.

Agora, sejam $a = (a_0, \ldots, a_{n-1}), b = (b_0, \ldots, b_{n-1}) \in \mathcal{C}_1$ e $c = (c_0, \ldots, c_{n-1}) \in \mathcal{C}_2$ tais que $d_H(a, b) = d_H$. Tome $w = (a_0, c_0, \ldots, a_{n-1}, c_{n-1})$ e $w' = (b_0, c_0, \ldots, b_{n-1}, c_{n-1})$ palavras de $\mathcal{C}$, então por (2.1), $d_H(w, w') = d_H(a, b) + d_H(c, c) = d_H(a, b) = d_H$. Além disso, para w e w', o número de índices consecutivos é $L = d_H$, pois todas as coordenadas correspondentes ao código $\mathcal{C}_2$ são iguais, e nenhuma das diferentes coordenadas de $\mathcal{C}_1$ estão em posições consecutivas. Logo, pelo Teorema 2.1.18

$$d_P(w, w') = d_H(w, w') + L = 2d_H,$$

e, portanto, $d_P(\mathcal{C}) = 2d_H$. □

Observação: No enunciado do Teorema 2.2.3, os códigos $\mathcal{C}_1$ e $\mathcal{C}_2$ não precisam necessariamente ter a mesma distância mínima de Hamming, pois pela demonstração do Teorema 2.2.3, se as distâncias mínima de Hamming destes códigos são d_1 e d_2, respectivamente, então $d_H(\mathcal{C}) = \min\{d_1, d_2\}$, resultado análogo ao Teorema 4.2.2.

Exemplo 2.2.4. *Considere os códigos* $\mathcal{C}_1 = \{0000, 1001, 1100\} \subset \mathbb{Z}_3$ *e* $\mathcal{C}_2 = \{2010, 0101\} \subset \mathbb{Z}_3$. *Então o código intercalado* $\mathcal{C}$ *obtido mesclando os códigos* $\mathcal{C}_1$ *e* $\mathcal{C}_2$ *é o código*

$$\mathcal{C} = \{02000100, 00010001, 12000110, 12100100, 10010011, 10110001\}$$

com comprimento $n = 8$, *número de palavras* $M = 2 \cdot 3 = 6$ *e* $d_H(\mathcal{C}) = d_H(\mathcal{C}_1) = d_H(\mathcal{C}_2) = 2$. *Além disso, fazendo* $u_1 = 02000100$, $u_2 = 00010001$ $u_3 = 12000110$, $u_4 = 12100100$, $u_5 = 10010011$ *e* $u_6 = 10110001$, *temos*

$$\begin{array}{lll} d_P(u_1, u_2) = 8 & d_P(u_1, u_3) = 4 & d_P(u_1, u_4) = 4 \\ d_P(u_1, u_5) = 8 & d_P(u_1, u_6) = 8 & d_P(u_2, u_3) = 8 \\ d_P(u_2, u_4) = 8 & d_P(u_2, u_5) = 4 & d_P(u_2, u_6) = 4 \\ d_P(u_3, u_4) = 4 & d_P(u_3, u_5) = 8 & d_P(u_3, u_6) = 8 \\ d_P(u_4, u_5) = 8 & d_P(u_4, u_6) = 8 & d_P(u_5, u_6) = 4. \end{array}$$

ou seja, $d_P(\mathcal{C}) = 2d_H(\mathcal{C}) = 4$.

2.3 Construção a partir de um Código Cíclico

Enquanto códigos intercalados possuem ótimas distâncias de pares em relação a suas distâncias de Hamming (fator 2), eles são geralmente inferiores aos códigos de pares construídos a partir de um código qualquer, até mesmo se os códigos utilizados para construir o código intercalado possuam boas distância de Hamming. Esse fato é comprovado no exemplo a seguir.

Exemplo 2.3.1. *Considere o (30,22)-código cíclico encurtado (descrito na Seção 6 do Capítulo 1) gerado pelo polinômio* $1 + x^2 + x^3 + x^8$*, cuja distância mínima de pares é* $d_P = 7$*. Pela Proposição 2.1.24, esse código pode corrigir 3 pares de erros, um a mais que o (30,22)-código obtido mesclando o (15,11)-código de Hamming (perfeito) com ele mesmo, cuja distância mínima de pares é apenas* $d_P = 6$*.*

O problema com a abordagem de intercalação é que ela otimiza a distância de pares dada a distância de Hamming, sem nenhuma tentativa de otimizar a distância de Hamming em si (os códigos intercalados são conhecidos por terem uma distância de Hamming fraca em relação ao seu comprimento). Assim, no restante dessa seção adotamos uma abordagem mais equilibrada de delimitação da distância de pares dada a distância de Hamming, usando inicialmente códigos que possuem melhores distâncias de Hamming, como os códigos lineares cíclicos. Para códigos na métrica de Hamming, os códigos mais poderosos, flexíveis e práticos em uso são os códigos cíclicos. Assim, o propósito dos próximos resultados é compreender como a estrutura dos códigos cíclicos também pode ser utilizada para a correção de erros de pares.

Proposição 2.3.2. *Seja* $c(x)$ *um polinômio em* $\frac{\mathbb{F}_q[x]}{<x^n-1>}$*, como* $\mathbb{F}_q$ *um corpo com* q *elementos, cujo DFT satisfaz* $\hat{C}_{b+1} = \hat{C}_{b+2} = \ldots = \hat{C}_{b+\delta} = 0$*, para algum inteiro* b*. Então o número de coeficientes não nulos em* $c(x)$ *é no mínimo* $\delta + 1$*.*

Demonstração. Como α é uma raiz n-ésima primitiva da unidade, $c \in \mathcal{C}$, com $c = \nu^{-1}(c(x))$ e

$$\mathcal{C} = \mathcal{C}_K(n, \delta) = \left\{ (a_0, \ldots, a_{n-1}) \in K^n; \sum_{i=0}^{n-1} a_i \alpha^{ij} = 0,\ j = b+1, \ldots, b+\delta \right\}$$

o código BCH definido pelo polinômio gerador

$$g(x) = mmc(m_{\alpha^{b+1}}(x), \ldots, m_{\alpha^{b+\delta}}(x)).$$

Agora, se fizermos $\delta' = \delta + 1$ e $a = b + 1$ no Teorema 2.3, temos

$$g(x) = mmc(m_{\alpha^a}(x), \ldots, m_{\alpha^{a+\delta'-2}}(x)).$$

Logo, $d_H \leq \delta' = \delta + 1$ e, portanto, $c(x)$ possui no mínimo $\delta + 1$ coeficientes não nulos. □

Proposição 2.3.3. *Seja $c(x)$ um polinômio em $\frac{\mathbb{F}_q[x]}{<x^n-1>}$ cujos coeficientes satisfazem $c_{b+1} = c_{b+2} = \ldots = c_{b+\delta} = 0$, para algum inteiro b. Então o número de coeficientes não nulos na sequência da DFT $\{\hat{C}_j\}_{j=0}^{n-1}$ é no mínimo $\delta + 1$.*

Demonstração. Como visto na Seção 1.8, temos

$$c(x) = \frac{1}{n} \sum_{i=0}^{n-1} \varphi_{\alpha,c}(\alpha^{-i}) x^i,$$

isto é, $c_j = \frac{1}{n}\varphi_{\alpha,c}(\alpha^{-j})$ para todo $i = 0, 1, \ldots, n-1$. Assim,

$$\varphi_{\alpha,c}(\alpha^{-(b+1)}) = \varphi_{\alpha,c}(\alpha^{-(b+2)}) = \ldots = \varphi_{\alpha,c}(\alpha^{-(b+\delta)}) = 0.$$

Logo, $\varphi_{\alpha,c}$ é um polinômio de $\frac{\mathbb{F}_q[x]}{<x^n-1>}$ que possui uma sequência de δ elementos consecutivos nulos da DFT , isto é, $\hat{C}'_{-(b+1)} = \hat{C}'_{-(b+2)} = \ldots = \hat{C}'_{-(b+\delta)}$, com $\hat{C}'_j = \varphi_{\alpha,c}(\alpha^j)$. Portanto, pela Proposição 2.3.2, $\varphi_{\alpha,c}$ possui no mínimo $\delta + 1$ coeficientes não nulos, ou seja, a sequência $\{\hat{C}_j\}_{j=0}^{n-1}$ possui no mínimo $\delta + 1$ elementos não nulos.

□

Lema 2.3.4. *Seja $\{\hat{C}_j\}_{j=0}^{n-1}$ uma sequência DFT com no mínimo d elementos nulos. Então $c(x)$ não tem um conjunto de $n-d$ coeficientes consecutivos tais que $c_{b+1} = c_{b+2} = \ldots = c_{b+n-d} = 0$.*

Demonstração. Suponha que $c(x)$ tenha um conjunto com $n-d$ coeficientes consecutivos nulos $c_{b+1} = \ldots = c_{b+n-d} = 0$. Então, pela Proposição 2.3.3, o número de elementos não nulos em $\{\hat{C}_j\}_{j=0}^{n-1}$ é no mínimo $n-d+1$, isto é, o número de elementos nulos em $\{\hat{C}\}_{j=0}^{n-1}$ é no máximo $n-(n-d+1) = d-1$, contradizendo a hipótese. Logo não existe um tal conjunto. □

Teorema 2.3.5. *Seja $g(x)$ um polinômio gerador de um código cíclico $\mathcal{C}$ com distância mínima de Hamming d_H. Se $g(x)$ tem no mínimo d_H raízes em $\mathbb{F}_{q^t}$, então a distância mínima de pares de $\mathcal{C}$ é no mínimo $d_H + 2$.*

Demonstração. Se $g(x)$ tem no mínimo d_H raízes, então qualquer palavra do código $c(x) = g(x)f(x)$ tem uma sequência de DFT com no mínimo d_H zeros. Pelo Lema 2.3.4, $c(x)$ não tem um conjunto de $n - d_H$ coeficientes nulos consecutivos. Vamos analisar dois casos:

- Se $c(x)$ tem peso exatamente d_H, então seus coeficientes não nulos podem cair em um único conjunto de coeficientes consecutivos, pois assim teríamos um conjunto de $n - d_H$ coeficientes nulos consecutivos, contradizendo a hipótese. O que implica, pelo Teorema 2.1.18, que o peso de pares é no mínimo $d_H + 2$.

- Agora, se o peso de $c(x)$ é estritamente maior que d_H, como cada coeficiente de $c(x)$ encontra-se em 2 coordenadas de vetores de pares e como $c(x)$ não possui um conjunto de $n - d_H$ coeficientes nulos consecutivos, segue que o peso de pares de $c(x)$ é no mínimo $d_H + 2$.

□

A importância do Teorema 2.3.5 é que fornece um limitante inferior algébrico $d_H(\mathcal{C})+2$ para a distância de pares de um código que é melhor que o limite inferior combinatorial $d_H(\mathcal{C})+1$ apresentado na Proposição 2.1.10. Pela Proposição 2.1.23, este limitante inferior algébrico aplica-se a todos os códigos cíclicos lineares que não são MDS. O próximo exemplo mostra como este limite inferior melhorado pode provar que códigos de Hamming cíclicos também são perfeitos na métrica de pares.

Exemplo 2.3.6. *Seja $\mathcal{C}$ um código cíclico de Hamming com comprimento $n = 2^t - 1$ gerado por $g(x)$, como visto na Seção 1.6. Vamos analisar a correção de erros de pares neste código. Para qualquer $t > 2$, $g(x)$ tem pelo menos $d_H = 3$ raízes. Assim, pelo Teorema 2.3.5, ele tem uma distância mínima de pares $d_P \geq 5$. Portanto, códigos de Hamming cíclicos de comprimento $n \geq 7$ podem corrigir 2 pares de erros. Mais adiante no Teorema 2.5.8, uma cota de Hamming para a métrica de pares será usado para provar que esses códigos de Hamming cíclicos são perfeitos também para a métrica de pares. Portanto, os códigos cíclicos de Hamming tem distância mínima de pares exatamente $d_P = 5$.*

Note que o limitante $d_P \geq 5$ obtido com o Teorema 2.3.5 no Exemplo 2.3.6 é usado exclusivamente para códigos de Hamming que são cíclicos. Por exemplo, existem formas equivalentes de construir códigos de Hamming, não cíclicos, com $d_P(\mathcal{C}) = 4$, como visto na Seção 1.4. Claramente, o código representado pela matriz teste de paridade $H_{[7,4]}$ abaixo é equivalente ao código gerado por $g(x)$ no Exemplo 2.3.6 (reordenando as coordenadas do código) e, como $(0001110) \in \mathcal{C}$, $d_P(\mathcal{C}) = 4$.

$$H_{[7,4]} = \begin{bmatrix} 1 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 \end{bmatrix}$$

No entanto, o fato de que os dois códigos tenham distâncias mínimas de pares diferentes demonstra a sensibilidade da correção de erro de pares para esse reordenamento de coordenadas.

A estrutura álgebrica dos códigos cíclicos serve à teoria da correção de pares de erros além do resultado do Teorema 2.3.5. Convenientemente, podemos aproveitar resultados mais profundos (do que o peso do código BCH) em códigos cíclicos para obter limites mais fortes na distância mínima de pares. Esta possibilidade será provada no Teorema 2.3.9.

Lema 2.3.7. *Seja $c(x)$ um polinômio em $\frac{\mathbb{F}_q[x]}{<x^n-1>}$ cuja DFT satisfaça $\hat{C}_{b+1} = \hat{C}_{b+2} = \ldots = \hat{C}_{b+\delta-1} = 0$ e $\hat{C}_{a+b+1} = \hat{C}_{a+b+2} = \ldots = \hat{C}_{a+b+\delta-1} = 0$, para inteiros b, a, com $mdc(a,n) = 1$ e $mdc(b,n) = 1$. Então o número de coeficientes não nulos em $c(x)$ é no mínimo $\delta + 1$.*

Demonstração. Por hipótese, temos que $c(x)$ pertence ao código

$$\mathcal{C} = I[g(x)] = \left\{ f(x) \in \frac{\mathbb{F}_q[x]}{< x^n - 1 >};\ f(\alpha^{b+i_1 a+i_2}) = 0, \text{ para todo } 0 \leq i_1 \leq 1 \text{ e } 1 \leq i_2 \leq \delta - 1 \right\},$$

com $\text{mdc}(a,n) = 1$ e $\text{mdc}(b,n) = 1$. Logo, pela generalização do peso estimado do código BCH, demonstrado no Teorema 1.6.8, o peso mínimo de Hamming do código $\mathcal{C}$ é no mínimo $\delta + 1$. Portanto, $\omega_H(c) \geq d_H(\mathcal{C}) \geq \delta + 1$. □

Lema 2.3.8. *Seja $\{\hat{C}_j\}_{j=0}^{n-1}$ uma sequência DFT com no mínimo m elementos nulos. Então $c(x)$ não tem dois conjuntos de $n - m - 1$ coeficientes consecutivos tais que $c_{b+1} = c_{b+2} = \ldots = c_{b+n-m-1} = 0$ e $c_{a+b+1} = c_{a+b+2} = \ldots = c_{a+b+n-m-1} = 0$, para qualquer b e qualquer a, com $\text{mdc}(a,n) = 1$ e $\text{mdc}(b,n) = 1$.*

Demonstração. Suponha que $c(x)$ tenha dois conjuntos tais que $c_{b+1} = c_{b+2} = \ldots = c_{b+n-m-1} = 0$ e $c_{a+b+1} = c_{a+b+2} = \ldots = c_{a+b+n-m-1} = 0$, com $\text{mdc}(a,n) < n - m$. Como visto na Seção 1.8, temos $c_j = \frac{1}{n}\varphi_{\alpha,c}(\alpha^{-j})$, para todo $i = 0, 1, \ldots, n-1$. Assim, $\varphi_{\alpha,c}(\alpha^{-(b+ia+c)}) = 0$, para todos $i = 0, 1$ e $c = 0, 1, \ldots, n-m-1$.

Logo, $\varphi_{\alpha,c}$ é um polinômio de $\frac{\mathbb{F}_q[x]}{<x^n-1>}$ que possui duas sequências de $n-m-1$ elementos consecutivos nulos da DFT , isto é, $\hat{C}'_{-(b+ia+c)} = 0$, com $\hat{C}_j = \varphi_{\alpha,c}(\alpha^j)$ e $i = 0, 1$ e $c = 0, 1, \ldots, n-m-1$. Assim, pela Proposição 2.3.2, $\varphi_{\alpha,c}$ possui no mínimo $n - m + 1$

coeficientes não nulos, ou seja, a sequência $\{\hat{C}_j\}_{j=0}^{n-1}$ possui no mínimo $n-m+1$ elementos não nulos. Daí, a sequência da DFT possui no máximo $n-(n-m+1) = m-1$ elementos nulos, contradizendo a hipótese, logo não existem tais sequências nulas. □

Teorema 2.3.9. *Seja $g(x)$ um polinômio gerador de um código cíclico $\mathcal{C}$ com comprimento primo n e distância mínima de Hamming d_H. Se $g(x)$ tem no mínimo m raízes em $\mathbb{F}_{q^t}$ e $d_H \leq \min(2m-n+2, m-1)$, então a distância mínima de pares de $\mathcal{C}$ é no mínimo d_H+3.*

Demonstração. Se $g(x)$ tem no mínimo m raízes em $\mathbb{F}_{q^t}$, então qualquer palavra do código $c(x) = g(x)f(x)$ tem uma sequência DFT com no mínimo m zeros. Pelo Lema 2.3.8, $c(x)$ não tem dois conjuntos de $n-m-1$ coeficientes nulos consecutivos, para qualquer espaçamento a entre eles (para n primo, $\text{mdc}(a,n) = 1$ e $\text{mdc}(b,n) = 1$, para qualquer $a < n$ e $b < n$).

Denote $D_H(c)$ o número de coeficientes não nulos em $c(x)$ e $D_o(c)$ o número de coeficientes nulos. Para $c(x)$ tal que $0 < \omega_H(c) < n$, uma das afirmações é verdadeira:

i) O número de coeficientes não nulos de $c(x)$ cai em um único subconjunto de índices consecutivos. Pelo Lema 2.3.4,

$$\begin{aligned} D_o(c) < n-m &\Rightarrow \omega_H(c) = n - D_o(c) > n - n + m = m \\ &\Rightarrow \omega_H(c) \geq m+1. \end{aligned}$$

Como $d_H \leq m-1$, então $m \geq d_H+1$, o que implica $\omega_H(c) \geq d_H+2$. Logo, pela Proposição 2.1.10, $\omega_P(c) \geq \omega_H(c)+1 \geq (d_H+2)+1 = d_H+3$.

ii) O número de coeficientes não nulos de $c(x)$ cai em apenas 2 subconjuntos de índices consecutivos. Como $c(x)$ não tem 2 conjuntos de $n-m-1$ coeficientes nulos consecutivos, temos $D_o(c) < 2(n-m-1)$. Logo, $\omega_H(c) = n - D_o(c) > n - 2(n-m-1) = 2m-n+2 \geq d_H$. Assim, $\omega_H(c) \geq d_H+1$ e, pelo Teorema 2.1.18,

$$\omega_P(c) = d_P(c,0) = d_H(c,0) + L = \omega_H(c) + 2 \geq d_H + 3,$$

com L o número de subconjuntos de índices consecutivos.

iii) O número de coeficientes não nulos de $c(x)$ cai em 3 ou mais subconjuntos consecutivos. Então, pelo Teorema 2.1.18, $\omega_P(c) = \omega_H(c) + L \geq \omega_H(c) + 3 \geq d_H + 3$. Portanto, a distância mínima de pares de $\mathcal{C}$ é no mínimo d_H+3.

□

2.4 A distância de pares de códigos cíclicos binários

Nessa seção será demonstrado que códigos cíclicos lineares binários fornecem uma maior distância mínima de pares com relação à distância de Hamming. Para isso, primeiro mostramos um método de determinar o peso de pares de um vetor $x \in \mathcal{A}^n$. Para os resultados dessa seção utilizaremos o alfabeto $\mathcal{A} = \{0, 1\}$.

A observação chave é que se $x_i = 1$, então dois símbolos no vetor de pares $\pi(x)$, o $(i-1)$-ésimo e i-ésimo símbolo de pares não serão nulos. É claro que a condição $x_{i-1} = 1$ também causa o $(i-1)$-ésimo par de símbolos ser não nulo. Assim, a medida que incrementamos o índice i, podemos pensar na condição $(x_{i-1}, x_i) = (0, 1)$ contribuindo com dois novos símbolos não nulos à $\pi(x)$, enquanto que a condição $(x_{i-1}, x_i) = (1, 1)$ contribui com apenas um. Assim, para determinar o peso de $\pi(x)$, é necessário determinar o número de ocorrências da sequência $(x_{i-1}, x_i) = (0, 1)$ no vetor x, que mostraremos como fazer a seguir.

Para $x = (x_0, x_1, \ldots, x_{n-1})$, definimos

$$x' = (x_0 + x_1, x_1 + x_2, \ldots, x_{n-1} + x_0). \tag{2.2}$$

Por exemplo, se $x = (00101101)$ então $x' = (01110111)$.

O próximo Lema fornece uma caracterização do peso de pares de um vetor x em função de x'.

Lema 2.4.1. *Para todo $x \in \mathcal{A}^n$, $\omega_P(x) = \omega_H(x) + \omega_H(x')/2$.*

Demonstração. Seja

$$S_0 = \{i;\ (x_i, x_{i+1}) \neq (0, 0) \text{ e } x_i = 1\},$$

$$S_1 = \{i;\ (x_i, x_{i+1}) \neq (0, 0) \text{ e } x_i = 0\}.$$

Assim, $\mid S_0 \mid = \omega_H(x)$, $S_0 \cap S_1 = \emptyset$, e $\omega_P(x) = \mid S_0 \mid + \mid S_1 \mid$. Para todo $0 \leq i \leq n-1$, $i \in S_1$ se, e somente se, $x_i = 0$ e $x_{i+1} = 1$. Neste caso, $x'_i = x_i + x_{i+1} = 1$. Portanto,

$$\mid S_1 \mid = \mid \{i;\ x_i = 0 \text{ e } x'_i = 1\} \mid .$$

Seja $S_2 = \{i;\ x_i = 1 \ e \ x'_i = 1\}$. Note que $\mid S_1 \mid = \mid S_2 \mid$. De fato, se $i \in S_2$, significa que i contribui em um à $\mid S_2 \mid$ e $(x_i, x_{i+1}) = (1, 0)$. Assim, $(x_{i+1}, x_{i+2}) = (0, *)$, com x_{i+2} valendo 0 ou 1. Se $x_{i+2} = 1$, então $(x_{i+1}, x_{i+2}) = (0, 1)$ e $i + 1 \in S_1$, isto é, $i + 1$ contribui com um à $\mid S_1 \mid$. Agora, se $x_{i+2} = 0$, então $(x_{i+2}, x_{i+3}) = (0, *)$, com $x_{i+3} = 0$ ou 1.

Continuando esse processo, encontraremos $j = 0, \ldots, n-1$ tal que $x_{i+j} = 1$, pois x é um vetor finito de comprimento n e $i, j \in \{0, 1, \ldots, n-1\}$. Assim, mesmo que $x_{i+j} = 0$, para todo $j = 1, \ldots, n-1$, temos $(x_{i+j-1}, x_{i+j}) = (0, 1)$, com $j = 0$. Logo, $i + j \in S_1$, isto é, contribui com um à $\mid S_1 \mid$. Portanto, $\mid S_2 \mid \leq \mid S_1 \mid$.

De forma análoga, verifica-se que para cada $i \in S_1$, somando 1 à $\mid S_1 \mid$, existe $i - k \in S_2$ somando 1 à $\mid S_2 \mid$, assim $\mid S_1 \mid \leq \mid S_2 \mid$.

Agora, observe que $\omega_H(x') = \mid S_2 \mid + \mid S_1 \mid = 2 \cdot \mid S_1 \mid$, assim $\mid S_1 \mid = \frac{\omega_H(x')}{2}$. Portanto,

$$\omega_P(x) = \mid S_0 \mid + \mid S_1 \mid = \omega_H(x) + \frac{\omega_H(x')}{2}.$$

□

Observação 2.4.2. *Uma consequência dessa demonstração é que $\omega_H(x')$ é sempre um inteiro positivo par, pois $\omega_H(x') = \mid S_2 \mid + \mid S_1 \mid = 2 \cdot \mid S_1 \mid$.*

Teorema 2.4.3. *Seja $\mathcal{C}$ um código cíclico linear de dimensão maior que 1. Então*

$$d_P(\mathcal{C}) \geq d_H(\mathcal{C}) + \left\lceil \frac{d_H(\mathcal{C})}{2} \right\rceil.$$

Demonstração. Seja $x = (x_0, \ldots, x_{n-1})$ uma palavra em $\mathcal{C}$. Suponha $x \neq \mathbf{1}$. Como o código é cíclico, $(x_1, \ldots, x_{n-1}, x_0) \in \mathcal{C}$ e como $\mathcal{C}$ é linear

$$x' = (x_0, \ldots, x_{n-1}) + (x_1, \ldots, x_{n-1}, x_0) \in \mathcal{C}.$$

Da Observação 2.4.2, o peso de x' é par e como $x \neq 1$, temos $x' \neq \mathbf{0}$. Daí, como $\omega_H(x') \geq d_H(\mathcal{C})$, temos $\frac{\omega_H(x')}{2} \geq \frac{d_H(\mathcal{C})}{2}$ e, como $\frac{\omega_H(x')}{2}$ é um inteiro não nulo, segue $\frac{\omega_H(x')}{2} \geq \left\lceil \frac{d_H(\mathcal{C})}{2} \right\rceil$. Além disso, $\omega_H(x) \geq d_H(\mathcal{C})$. Assim,

$$\omega_P(x) = \omega_H(x) + \omega_H(x')/2 \geq d_H(\mathcal{C}) + \left\lceil \frac{d_H(\mathcal{C})}{2} \right\rceil.$$

Agora, se $x = \mathbf{1}$ é uma palavra do código $\mathcal{C}$, então $\omega_P(x) = n$. Afirmamos que se a dimensão de $\mathcal{C}$ é maior que um, então $d_H(\mathcal{C}) \leq \lfloor 2n/3 \rfloor$. Isto implica $d_H(\mathcal{C}) \leq \lfloor 2n/3 \rfloor \leq 2n/3$, daí $\frac{d_H(\mathcal{C})}{2} \leq \frac{n}{3}$ e, como a função teto é crescente, temos

$$d_H(\mathcal{C}) + \left\lceil \frac{d_H(\mathcal{C})}{2} \right\rceil \leq \lfloor 2n/3 \rfloor + \lceil n/3 \rceil < \frac{2n}{3} + \frac{n}{3} + 1 = n + 1,$$

pois $\lceil \frac{n}{3} \rceil < \frac{n}{3} + 1$. Logo,

$$d_H(\mathcal{C}) + \left\lceil \frac{d_H(\mathcal{C})}{2} \right\rceil \leq n = \omega_P(x).$$

Para provar a afirmação, suponha, por contradição, $d_H(\mathcal{C}) > \lfloor \frac{2n}{3} \rfloor$, isto é, $d_H(\mathcal{C}) \geq \lfloor \frac{2n}{3} \rfloor + 1$. Como $\dim(\mathcal{C}) \geq 2$, existem $u, v \in \mathcal{C}$ não nulos e linearmente independentes, portanto, distintos. Então $\omega_H(u),\ \omega_H(v) \geq d_H(\mathcal{C}) \geq \lceil 2n/3 \rceil + 1 > 2n/3$.

Seja $\omega_0(x) :=\mid \{i;\ x_i = 0\} \mid$, então

$$\omega_0(u), \omega_0(v) \leq n - 2n/3 = n/3.$$

O fato $u_i + v_i \neq 0$ implica $u_i \neq v_i$, isto é, $u_i + v_i = 1$ implica ou $u_i = 0$ e $v_i = 1$, ou $u_i = 1$ e $v_i = 0$. Daí, $\omega_H(u+v) \leq \omega_0(u) + \omega_0(v) \leq \frac{2n}{3}$. Como $\omega_H(u+v) \in \mathbb{Z}$, $\omega_H(u+v) \leq \lfloor 2n/3 \rfloor$ o que é uma contradição, pois como $\mathcal{C}$ é linear, $u + v \in \mathcal{C}$ e deveríamos ter $\omega_H(u+v) \geq d_H(\mathcal{C}) > \lfloor 2n/3 \rfloor$. Portanto, $d_H(\mathcal{C}) \leq \lfloor 2n/3 \rfloor$, completando a demonstração. □

2.5 Limitantes do Tamanho dos Códigos

2.5.1 Limites Combinatoriais

A existência de condições necessárias e suficientes para a correção dos erros de pares permite a derivação dos limitantes superior e inferior, respectivamente, do tamanho do código. Uma técnica bem conhecida, usada para ambos os tipos de limitantes, é contar o número de palavras à distância d de uma determinada palavra. Na métrica de Hamming, essa tarefa de contagem é muito simples e é usada para derivar a cota de Hamming (superior) e a cota de Gilbert-Varshamov (inferior), dentre muitos outros limitantes [13] [19]. Dada uma palavra de $\mathcal{A}^n$, na métrica de pares há a complicação de termos parte dos vetores de pares com erros (os consistentes) resultando em palavras de $\mathcal{A}^n$, enquanto outros vetores de pares com erros corresponde a vetores de pares não consistentes. Assim, o desafio é contar apenas os vetores de pares consistentes à distância d_P de uma palavra dada.

Pelo Teorema 2.1.18, temos $h = d_P(x, y) = \sum_{i=1}^{L} \mid B_i \mid + L$. Fazendo $\ell = \sum_{i=1}^{L} \mid B_i \mid$ temos $L = h - \ell$. Assim, esse problema pode ser reescrito como quantos dos subconjuntos do conjunto de coordenadas $[0, n-1]$ têm tamanho total ℓ e partição mínima de subconjuntos consecutivos $L = h - \ell$ (ciclicamente). Se este problema for resolvido, todas

as palavras que diferem da palavra dada nestes subconjuntos de tamanho ℓ são as que estão à distância de pares h desta palavra. Seja $D(n, \ell, L)$ o número de subconjuntos de $[0, n-1]$ com tamanho total ℓ que ocupam L subconjuntos consecutivos ciclicamente.

Teorema 2.5.1. *Para* $L \leq \ell < n$,

$$D(n, \ell, L) = \frac{n}{L} \begin{pmatrix} \ell - 1 \\ L - 1 \end{pmatrix} \begin{pmatrix} n - \ell - 1 \\ L - 1 \end{pmatrix}$$

Demonstração. Um subconjunto que atende à especificação (n, ℓ, L) possui uma das formas representados na Figura 2.1. Os retângulos escuros representam elementos no subconjunto de tamanho ℓ. Os retângulos brancos representam elementos que não estão no subconjunto.

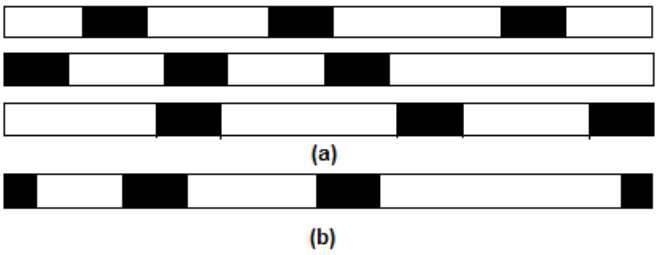


Figura 2.1: *Layouts* de subconjuntos (n, ℓ, L). (a)*Layouts* sem contorno. (b) *Layouts* com contorno.

Os três *layouts* em (a) são aqueles que não têm nenhum contorno de $n-1$ para 0. O layout em (b) tem um subconjunto consecutivo que contorna. Para qualquer L, os retângulos escuros e brancos são colocados alternadamente (a Figura 2.1 apresenta um exemplo com $L = 3$). Cada retângulo representa um subconjunto com tamanho estritamente maior que 0. Os tamanhos dos retângulos escuros em cada layout somam ℓ. Os tamanhos dos retângulos brancos somam $n - \ell$. Considere os índices brancos / escuros como elementos idênticos e os retângulos brancos / escuros de índices consecutivos como as caixas. Da contagem elementar, o número de maneiras de dividir m elementos idênticos em k caixas numeradas não vazias é $\begin{pmatrix} m-1 \\ k-1 \end{pmatrix}$, ([27], Cap. 13).

Observamos que os elementos escuros e brancos podem ser agrupados de forma independente em retângulos, e cada um desses agrupamentos fornece um subconjunto distinto (n, ℓ, L). Portanto, o número de subconjuntos (n, ℓ, L) de cada layout é produto do número de agrupamentos escuros pelo número de agrupamentos claros. Em particular, os 4 *layouts* na Figura 2.1 tem os seguintes parâmetros m, k, listados de cima para baixo.

1. Escuros: $m = \ell$, $k = L$. Brancos: $m = n - \ell$, $k = L + 1$

2. Escuros: $m = \ell$, $k = L$. Brancos: $m = n - \ell$, $k = L$

3. Escuros: $m = \ell$, $k = L$. Brancos: $m = n - \ell$, $k = L$

4. Escuros: $m = \ell$, $k = L + 1$. Brancos: $m = n - \ell$, $k = L$

Assim, somando o número de subconjuntos de cada layout, temos

$$\begin{aligned} D(n,\ell,L) &= \begin{pmatrix} \ell-1 \\ L-1 \end{pmatrix}\begin{pmatrix} n-\ell-1 \\ L \end{pmatrix} + 2\begin{pmatrix} \ell-1 \\ L-1 \end{pmatrix}\begin{pmatrix} n-\ell-1 \\ L-1 \end{pmatrix} && (2.3)\\ &+ \begin{pmatrix} n-\ell-1 \\ L-1 \end{pmatrix}\begin{pmatrix} \ell-1 \\ L \end{pmatrix} && (2.4)\\ &= \frac{n}{L}\begin{pmatrix} \ell-1 \\ L-1 \end{pmatrix}\begin{pmatrix} n-\ell-1 \\ L-1 \end{pmatrix}. && (2.5) \end{aligned}$$

□

Observação 2.5.2. *Vale a pena notar casos especiais interessantes de $D(n,\ell,L)$ do Teorema 2.5.1.*

- *$D(n,\ell,1) = n$ (um conjunto único com deslocamento arbitrário).*
- *$D(n,\ell,L) = 0$, se $L > \ell$ ou $L > n - \ell$.*
- *Para $D(n,\ell,\ell)$, a contribuição de (2.4) é zero, pois cada um dos ℓ subconjuntos possuem apenas um elemento cada, ou seja, as coordenadas 0 e $n-1$ não podem pertencer ao mesmo subconjunto.*

Com uma fórmula aproximada para $D(n,\ell,L)$ é possível obter uma fórmula aproximada para o número de palavras de $\mathcal{A}^n$ à distância de pares h de uma palavra dada. Para isso, definimos $S_h(x)$, a *esfera de pares* de raio h em torno de uma palavra x.

Definição 2.5.3. *Para uma palavra $x \in \mathcal{A}^n$, defina a* ***esfera de pares*** *$S_h(x)$ como o conjunto de todos $y \in \mathcal{A}^n$ tais que $d_P(x,y) = h$ e o* **disco de pares** *ou* **bola de pares** *$B_h(x)$ consiste em todas as palavras com distância de pares $\leq h$ de x.*

O tamanho da esfera de raio h é dada na seguinte proposição.

Proposição 2.5.4. *Para qualquer $x \in \mathcal{A}^n$ e $0 < h < n$,*

$$\mid S_h(x) \mid = \sum_{\ell=\lceil h/2 \rceil}^{h-1} D(n,\ell,h-\ell)(q-1)^\ell,$$

com $q = \mid \mathcal{A} \mid$ o tamanho do alfabeto $\mathcal{A}$.

Demonstração. O número de palavras à distância de pares h de x é o número de possíveis combinações dos vetores (cada vetor de pares tem h coordenadas não nulas com n possibilidades de posições diferentes) multiplicado pelo número de possibilidades de x em $\mathcal{A}$ (cada posição pode conter $q-1$ letras diferentes de $\mathcal{A}$). Assim,

$$\mid S_h(x) \mid = \sum_{\ell=1}^{h} D(n,\ell,L)(q-1)^{\ell} = \sum_{\ell=1}^{h} D(n,\ell,h-\ell)(q-1)^{\ell}.$$

Note que se $\ell = h$, então $L = 0$, pois $h = \ell + L$, assim $D(n,h,h-\ell) = 0$. Se $\ell < h/2$, então $2\ell < h = \ell + L$, ou seja, $L > \ell$, logo $D(n,\ell,L) = 0$. Como $\ell \in \mathbb{N}$, temos

$$\mid S_h(x) \mid = \sum_{\ell=\lceil h/2 \rceil}^{h-1} D(n,\ell,h-\ell)(q-1)^{\ell}.$$

□

Note que $\mid S_1(x) \mid = 0$ e $\mid S_2(x) \mid = n(q-1)$, que coincide com a esfera de Hamming de raio 1. Os casos extremos $h = 0$ e $h = n$ são $\mid S_0(x) \mid = 1$ e $\mid S_n(x) \mid = (q-1)^n$, respectivamente, idêntico à esfera da métrica de Hamming com o mesmo raio. Além disso,

$$\mid B_h(x) \mid = 1 + \sum_{i=1}^{h} \mid S_i(x) \mid . \tag{2.6}$$

Observe ainda que o número de elementos de $S_h(x)$ e de $B_h(x)$ depende apenas de n, q e h. Logo, $\mid S_h(x) \mid = \mid S_h(y) \mid$ e $\mid B_h(x) \mid = \mid B_h(y) \mid$, para todos $x, y \in \mathcal{A}^n$.

Lema 2.5.5. *Seja $\mathcal{C}$ um código com distância mínima d_P que corrige t-pares de erros. Se c e c' são palavras distintas de $\mathcal{C}$, então*

$$B_t(c) \cap B_t(c') = \emptyset.$$

Demonstração. Pela Proposição 2.1.24, $\mathcal{C}$ corrige t-pares de erros se, e somente se, $d_P \geq 2t+1$, isto é, $t \leq \frac{d_P - 1}{2}$. Se existe $x \in B_t(c) \cap B_t(c')$ então $d_P(c,x) \leq t$ e $d_P(x,c') \leq t$. Assim, pela desigualdade triangular temos

$$d_P(c,c') \leq d_P(c,x) + d_P(x,c') \leq 2t \leq d_P - 1,$$

o que contradiz $d_P(c,c') \geq d_P$. Portanto, $B_t(c) \cap B_t(c') = \emptyset$. □

Proposição 2.5.6 (Cota de Hamming para Métrica de Pares)**.** *Se $\mathcal{C} \subset \mathcal{A}^n$ é um código com M palavras que corrige todos t pares de erros, então*

$$M \mid B_t(x) \mid \leq q^n.$$

Demonstração. Se $\mathcal{C}$ é um código que corrige t-pares de erros, então, pelo Lema 2.5.5 se $c, c' \in \mathcal{C}$,

$$B_t(c) \cap B_t(c') = \emptyset$$

e como

$$\bigcup_{c \in \mathcal{C}} B_t(c) \subset \mathcal{A}^n,$$

segue

$$q^n \geq \sum_{c \in \mathcal{C}} \mid B_t(c) \mid = M \mid B_t(x) \mid .$$

□

Definição 2.5.7. *Os códigos que satisfazem a Proposição 2.5.6 com igualdade são chamados de* **códigos perfeitos na métrica de pares**.

Esta cota inferior na métrica de pares pode ser usada para provar que os códigos cíclicos de Hamming (Seção 1.5) analizados no Exemplo 2.3.6 também são perfeitos na métrica de pares.

Teorema 2.5.8. *Se $\mathcal{C}$ é um código perfeito na métrica de Hamming com $d_H = 3$ e, além disso, tem distância mínima de pares $d_P = 5$, então $\mathcal{C}$ também é perfeito na métrica de pares.*

Demonstração. Como $d_P = 5$, $t = \left\lfloor \frac{d_P - 1}{2} \right\rfloor = 2$. Dado $c \in \mathbb{F}_2^n$, temos

$$\mid B_2(c) \mid = 1 + n(q-1) = |D(c,1)|,$$

ou seja, o número de elementos em uma bola de raio 2 na métrica de pares é o mesmo número de elementos em uma bola de raio 1 na métrica de Hamming (Lema 1.1.6). Por hipótese, $\mathcal{C}$ é perfeito na métrica se Hamming, ou seja,

$$\bigcup_{c \in \mathcal{C}} D(c,1) = \mathbb{F}_q^n,$$

assim,

$$q^n = \mid \bigcup_{c \in \mathcal{C}} D(c,1) \mid = \mid \bigcup_{c \in \mathcal{C}} B_2(c) \mid$$

Logo,

$$\bigcup_{c\in\mathcal{C}} B_2(c) = \mathbb{F}_q^n.$$

□

O Teorema 2.5.8 implica que códigos de Hamming com $d_P = 5$ também são perfeitos na métrica de pares, assim não podem ter distância de pares $d_P = 6$, mesmo essa distância satisfazendo $d_P \leq 2d_H$. É importante notar que esse Teorema é para o caso específico $(d_H, d_P) = (3, 5)$, em geral, códigos perfeitos na métrica de Hamming podem não ser perfeitos na métrica de pares, o mesmo vale para o caso contrário.

Agora, descrevemos o limite inferior de Gilbert-Varshamov para códigos de pares.

Proposição 2.5.9 (Cota de Gilbert-Varshamov)**.** *Existe um código $\mathcal{C} \in \mathcal{A}^n$ com M palavras e distância mínima de pares d_P com*

$$M \mid B_{d-1}(x) \mid \geq q^n.$$

Demonstração. Considere a seguinte função

$$A(n, d_P) = \max\{M; \text{existe um código com parâmetros}[n, M, d_P]\}. \tag{2.7}$$

Seja $\mathcal{C}$ o código com $M = A(n, d_P)$ palavras, comprimento n e distância mínima de pares d_P. Temos

$$\bigcup_{c\in\mathcal{C}} B_{d_P-1}(c) = \mathcal{A}^n,$$

pois se existisse $c' \in \mathcal{A}^n \setminus \bigcup_{c\in\mathcal{C}} B_{d_P-1}(c)$, teríamos que $\hat{\mathcal{C}}' = \mathcal{C} \cup \{c'\}$ seria um código com parâmetros $[n, M + 1, d_P]$, contradizendo (2.7). Logo,

$$q^n = \mid \mathcal{A}^n \mid = \mid \bigcup_{c\in\mathcal{C}} B_{d_P-1}(c) \mid \leq \sum_{c\in\mathcal{C}} \mid B_{d_P-1}(c) \mid = M \mid B_{d_P-1}(x) \mid .$$

□

2.5.2 Limites Assintóticos

Os limites combinatórios da seção anterior usam uma numeração exata de esferas de pares e são portanto uma ferramenta útil para limitar o tamanho dos códigos com parâmetros determinados. No entanto, para obter uma visão geral sobre a viabilidade e limites na codificação no modelo de pares de erros, uma análise assintótica é necessária.

A principal tarefa em direção a uma análise assintótica é derivar limites concisos nos tamanhos das bolas de pares. Em seguida, as expressões simples resultantes são usadas para ligar as taxas de códigos com a distância mínima de pares fracionada $\delta = d_P/n$ (com o comprimento do código n tendendo para o infinito).

Nosso objetivo é obter limites assintóticos sobre o tamanho das bolas de pares que serão o suficiente para mostrar uma vantagem na taxa de codificação no esquema de pares sobre código no esquema de Hamming. Observamos inicialmente que não é claro que tal vantagem exista. Considerando a desigualdade $d_H + 1 \leq d_P \leq 2d_H$, se os códigos de pares assintoticamente bons tiverem distância de pares baixas mais próximas à $d_H + 1$, então eles provavelmente não terão qualquer vantagem sobre os códigos na métrica de Hamming. Por outro lado, se os códigos de pares assintoticamente bons tiverem distância de pares altas, próximas à $2d_H$, então uma vantagem significativa surgirá a favor dos códigos de pares. Assim, o principal objetivo da análise abaixo é ver se os códigos de pares assintoticamente bons caem na extremidade inferior, na superior ou em algum lugar entre eles (possui uma vantagem assintótica porém menor do que duplicar distância relativa).

Começamos por obter um limite superior simples para $D(n, \ell, L)$ pela seguinte desigualdade

$$\begin{aligned} D(n,\ell,L) &= \binom{\ell-1}{L-1}\binom{n-\ell-1}{L} + 2\binom{\ell-1}{L-1}\binom{n-\ell-1}{L-1} \\ &+ \binom{\ell-1}{L}\binom{n-\ell-1}{L-1} \\ &< 4\binom{\ell}{L}\binom{n-\ell}{L}. \end{aligned} \tag{2.8}$$

A desigualdade decorre da recursão binomial básica que fornece as seguintes desigualdades

$$\binom{a-1}{b} = \binom{a}{b} - \binom{a-1}{b-1} < \binom{a}{b} \tag{2.9}$$

e

$$\binom{a-1}{b-1} = \binom{a}{b} - \binom{a-1}{b} < \binom{a}{b}$$

(substitua $b = L$ e $a = l$ ou $a = n - \ell$ para conseguir (2.8)). O tamanho de uma esfera de pares agora pode ser limitado usando (2.8), como segue.

$$\mid S_h(x) \mid = \sum_{\ell=\lceil h/2 \rceil}^{h-1} D(n,\ell,h-\ell)(q-1)^\ell < 4 \sum_{\ell=\lceil h/2 \rceil}^{h-1} \binom{\ell}{h-\ell}\binom{n-\ell}{h-\ell}(q-1)^\ell.$$

Substituindo em (2.6), temos

$$\mid B_h(x) \mid = 1 + \sum_{i=1}^{h} \mid S_i(x) \mid \leq 4 \sum_{i=1}^{h} \sum_{\ell=\lceil i/2 \rceil}^{i-1} \left(\begin{array}{c} \ell \\ i-\ell \end{array} \right) \left(\begin{array}{c} n-\ell \\ i-\ell \end{array} \right) (q-1)^{\ell}.$$

Como $\left(\begin{array}{c} n-\ell \\ i-\ell \end{array} \right) = \left(\begin{array}{c} n-\ell \\ n-i \end{array} \right)$, temos

$$\mid B_h(x) \mid \leq 4 \sum_{i=1}^{h} \sum_{\ell=\lceil i/2 \rceil}^{i-1} \left(\begin{array}{c} \ell \\ i-\ell \end{array} \right) \left(\begin{array}{c} n-\ell \\ n-i \end{array} \right) (q-1)^{\ell}. \tag{2.10}$$

Observe que, para $u, v, w \in \mathbb{N}$, se $u \geq v$, então $u = v + a$, com $a \geq 0$, então por (2.9),

$$\left(\begin{array}{c} u \\ w \end{array} \right) > \left(\begin{array}{c} u-1 \\ w \end{array} \right) > \ldots > \left(\begin{array}{c} u-a \\ w \end{array} \right) = \left(\begin{array}{c} v \\ w \end{array} \right) \tag{2.11}$$

Daí, como $\ell \geq \lceil i/2 \rceil$, $n - \ell \leq n - \lceil i/2 \rceil$, então podemos mover o segundo multiplicando para fora da soma interna em (2.10), obtendo

$$4 \sum_{i=1}^{h} \sum_{\ell=\lceil i/2 \rceil}^{i-1} \left(\begin{array}{c} \ell \\ i-\ell \end{array} \right) \left(\begin{array}{c} n-\ell \\ n-i \end{array} \right) (q-1)^{\ell} <$$

$$\underbrace{4 \sum_{i=1}^{h} \left(\begin{array}{c} n-\lceil i/2 \rceil \\ n-i \end{array} \right)}_{(1)} \underbrace{\sum_{\ell=\lceil h/2 \rceil}^{i-1} \left(\begin{array}{c} \ell \\ i-\ell \end{array} \right) (q-1)^{\ell}}_{(2)}. \tag{2.12}$$

Vamos limitar o somatório (1). Fazendo $k = \ell - \lfloor i/2 \rfloor$, temos $i - \ell = i - \lceil i/2 \rceil - k = i + \lfloor -i/2 \rfloor - k = \lfloor i/2 \rfloor - k$ e quando $\ell = i-1$, $k = i-1-\lceil i/2 \rceil = i-1+\lfloor -i/2 \rfloor = \lfloor i/2 \rfloor - 1$, assim

$$\sum_{l=\lceil i/2 \rceil}^{i-1} \left(\begin{array}{c} \ell \\ i-\ell \end{array} \right) (q-1)^{\ell} = \sum_{k=0}^{\lfloor i/2 \rfloor - 1} \left(\begin{array}{c} \lceil i/2 \rceil + k \\ \lfloor i/2 \rfloor - k \end{array} \right) (q-1)^{k+\lfloor i/2 \rfloor}.$$

Como $\left(\begin{array}{c} a \\ b \end{array} \right) = \left(\begin{array}{c} a \\ a-b \end{array} \right)$, temos

$$\sum_{k=0}^{\lfloor i/2 \rfloor - 1} \left(\begin{array}{c} \lceil i/2 \rceil + k \\ \lfloor i/2 \rfloor - k \end{array} \right) (q-1)^{k+\lfloor i/2 \rfloor} = \sum_{k=0}^{\lfloor i/2 \rfloor - 1} \left(\begin{array}{c} \lceil i/2 \rceil + k \\ 2k + i (\mathrm{mod}\ 2) \end{array} \right) (q-1)^{k+\lfloor i/2 \rfloor},$$

pois se i é par, então $\lceil i/2 \rceil - \lfloor i/2 \rfloor = i/2 - i/2 = 0$ e, se i é ímpar, então $i = 2s + 1$, daí

$\lceil i/2\rceil - \lfloor i/2\rfloor = \lceil \frac{2s+1}{2}\rceil - \lfloor \frac{2s+1}{2}\rfloor = s+1-s = 1$, logo $\lceil i/2\rceil + \lfloor i/2\rfloor = i(\mathrm{mod}\,2)$.

Agora, como $k \leq \lfloor i/2\rfloor - 1$ temos, $k + \lceil i/2\rceil \leq \lceil i/2\rceil + \lfloor i/2\rfloor - 1 = i-1$ e por (2.11),

$$\sum_{k=0}^{\lfloor i/2\rfloor -1} \left(\begin{array}{c} \lceil i/2\rceil + k \\ 2k + i(\mathrm{mod}\ 2) \end{array} \right) (q-1)^{k+\lfloor i/2\rfloor} < \sum_{k=0}^{\lfloor i/2\rfloor -1} \left(\begin{array}{c} i-1 \\ 2k + i(\mathrm{mod}\ 2) \end{array} \right) (q-1)^{i-1}.$$

Note que, $0 = i(\mathrm{mod}\,2)$ ou $1 = i(\mathrm{mod}\,2)$, daí por (2.9) temos

$$\left(\begin{array}{c} i-1 \\ 2k + i(\mathrm{mod}\ 2) \end{array} \right) = \left(\begin{array}{c} i-1 \\ 2k+1 \end{array} \right) < \left(\begin{array}{c} i \\ 2k+1 \end{array} \right) \text{ ou}$$

$$\left(\begin{array}{c} i-1 \\ 2k + i(\mathrm{mod}\ 2) \end{array} \right) = \left(\begin{array}{c} i-1 \\ 2k \end{array} \right) < \left(\begin{array}{c} i \\ 2k+1 \end{array} \right), \text{ logo}$$

$$\sum_{k=0}^{\lfloor i/2\rfloor -1} \left(\begin{array}{c} i-1 \\ 2k + i(\mathrm{mod}\ 2) \end{array} \right) (q-1)^{i-1} < \sum_{k=0}^{\lfloor i/2\rfloor -1} \left(\begin{array}{c} i \\ 2k+1 \end{array} \right) (q-1)^{i}.$$

Fazendo $j = 2k+1$, como $0 \leq k \leq \lfloor i/2\rfloor - 1$, temos $1 \leq j \leq 2\lfloor i/2\rfloor - 2 + 1 \leq i-1$, isto é,

$$\sum_{k=0}^{\lfloor i/2\rfloor -1} \left(\begin{array}{c} i \\ 2k+1 \end{array} \right) (q-1)^{i} < \sum_{j=1}^{i-1} \left(\begin{array}{c} i \\ j \end{array} \right) (q-1)^{i} < \sum_{j=0}^{i} \left(\begin{array}{c} i \\ j \end{array} \right) (q-1)^{i} = q^i \leq q^h \quad (2.13)$$

Considerando o somatório (2), reescrevemos, usando

$$\left(\begin{array}{c} n - \lceil i/2\rceil \\ n-i \end{array} \right) = \left(\begin{array}{c} n - \lceil i/2\rceil \\ n - \lceil i/2\rceil - (n-i) \end{array} \right) = \left(\begin{array}{c} n - \lceil i/2\rceil \\ i + \lfloor -i/2\rfloor \end{array} \right) = \left(\begin{array}{c} n - \lceil i/2\rceil \\ \lfloor i/2\rfloor \end{array} \right),$$

e continuamos com a seguinte cadeia de inequações

$$\sum_{i=1}^{h} \left(\begin{array}{c} n - \lceil i/2\rceil \\ \lfloor i/2\rfloor \end{array} \right) \overset{(2.11)}{<} \sum_{i=1}^{h} \left(\begin{array}{c} n \\ \lfloor i/2\rfloor \end{array} \right),$$

fazendo $t = \lfloor i/2\rfloor$, como $1 \leq i \leq h$, $0 = \lfloor 1/2\rfloor \leq t \leq \lfloor h/2\rfloor$, assim

$$\sum_{i=1}^{h} \left(\begin{array}{c} n \\ \lfloor i/2\rfloor \end{array} \right) < \sum_{t=0}^{\lfloor h/2\rfloor} \left(\begin{array}{c} n \\ t \end{array} \right) < q \sum_{t=0}^{\lfloor h/2\rfloor} \left(\begin{array}{c} n \\ t \end{array} \right)$$

Pelo Lema 1.7.10,

$$n \cdot H_q\left(\frac{h}{2n}\right) \geq \log_q(V_q(n, \lfloor h/2 \rfloor)) = \log_q \sum_{t=0}^{\lfloor h/2 \rfloor} \left(\begin{array}{c} n \\ t \end{array} \right) (q-1)^t$$

assim,

$$q^{n \cdot H_q\left(\frac{h}{2n}\right)} \geq \sum_{t=0}^{\lfloor t/2 \rfloor} \left(\begin{array}{c} n \\ t \end{array} \right) (q-1)^t,$$

logo, como $q \geq 2$

$$q \sum_{t=0}^{\lfloor t/2 \rfloor} \left(\begin{array}{c} n \\ t \end{array} \right) \leq q \sum_{t=0}^{\lfloor t/2 \rfloor} \left(\begin{array}{c} n \\ t \end{array} \right) (q-1)^t \leq q^{1+n \cdot H_q\left(\frac{h}{2n}\right)}$$

com $H_q(\alpha)$ a função entropia (Definição 1.7.9). Portanto,

$$\sum_{i=1}^{h} \left(\begin{array}{c} n - \lceil i/2 \rceil \\ \lfloor i/2 \rfloor \end{array} \right) < q^{1+n \cdot H\left(\frac{h}{2n}\right)}. \tag{2.14}$$

Combinando (2.13) e (2.14) temos

$$\mid B_h(x) \mid < 4 \cdot q^{1+nH\left(\frac{h}{2n}\right)+h}. \tag{2.15}$$

Denotando $\delta = h/n$ como a taxa de correção de erros, temos

$$\mid B_h(x) \mid < 4 \cdot q^{1+n\left[H\left(\frac{\delta}{2}\right)+\delta\right]}.$$

Para obter a cota assintótica na cota de um código é útil limitar a relação $q^n / \mid B_h(x) \mid$

$$\frac{q^n}{\mid B_h(x) \mid} > \frac{1}{4} \cdot q^{n-n\left[H\left(\frac{\delta}{2}\right)+\delta\right]-1}.$$

Assim, pela Cota de Gilber-Varshamov (Proposição 2.5.9)

$$A_q(n, \delta n) \geq \frac{q^n}{\mid B_{h-1}(x) \mid} \geq \frac{q^n}{\mid B_h(x) \mid} > \frac{1}{4} \cdot q^{n-n\left[H\left(\frac{\delta}{2}\right)+\delta\right]-3}.$$

Logo,

$$R_q(\delta) = \lim_{n \to \infty} \sup \frac{1}{n} \log_q(A_q(n, \delta n)) > 1 - H_q\left(\frac{\delta}{2}\right) - \delta. \tag{2.16}$$

Finalmente podemos escrever a cota inferior assintótica de Gilbert-Varshamov para

pares de erros.

Corolário 2.5.10 (Cota Assintótica de Gilbert-Varshamov)**.** *Existem códigos com distância mínima de pares fracionária δ e taxa*

$$R_q(\delta) > 1 - H_q\left(\frac{\delta}{2}\right) - \delta.$$

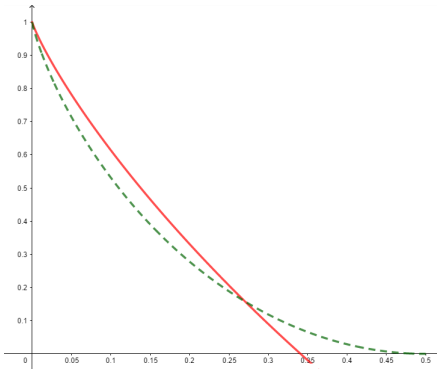


Figura 2.2: Comparação entre a cota assintótica de Gilbert-Varshamov para distância de pares (linha contínua) e para distância de Hamming (linha pontilhada), com $q = 2$.

A Figura 2.2 implica que os códigos de pares de erros existem com taxas assintóticas que são estritamente superiores as taxas de códigos comprovadamente realizáveis para erros de símbolos. Esta declaração se aplica a todas as distâncias fracionadas $\delta < 0.27$ (a coordenada x do ponto de interseção entre as duas curvas na figura).

A desigualdade em (2.16) não pode ser usada para excluir a existência de códigos com taxa mais alta (uma desigualdade reversa é necessária para isso). No entanto, é possível usar a cota assintótica para obter um limite inferior na cota de Hamming para pares de erros. A utilidade de tal limite inferior é expressa na seguinte proposição.

Proposição 2.5.11 (Cota no Tamanho de Códigos Perfeitos)**.** *Se existem códigos perfeitos na métrica de pares, com distância mínima de pares fracionária δ, então suas taxas assintóticas satisfazem*

$$R(\delta) > 1 - H_q\left(\frac{\delta}{4}\right) - \frac{\delta}{2}.$$

Demonstração. Seja $\mathcal{C}$ um código perfeito na métrica de pares com distância mínima de pares fracionária δ, então $M \cdot \mid B_{\frac{\delta n-1}{2}}(x) \mid = q^n$. Substituindo este valor em (2.15),

$$\mid B_{\frac{\delta n-1}{2}}(x) \mid < q^{1+n\left[H_q\left(\frac{\delta n-1}{4n}\right)+\frac{\delta n-1}{2n}\right]},$$

assim

$$\frac{q^n}{\mid B_{\frac{\delta n-1}{2}}(x) \mid} > q^{n-n\left[H_q\left(\frac{\delta}{4}-\frac{1}{4n}\right)+\frac{\delta}{2}-\frac{1}{2n}\right]-1}.$$

Portanto,

$$\begin{aligned} R_q(\delta) &= \limsup_{n\to\infty} n^{-1} \log_q \frac{q^n}{\mid B_{\frac{\delta n-1}{2}}(x) \mid} > \limsup_{n\to\infty} 1 - H_q\left(\frac{\delta}{4} - \frac{1}{4n}\right) - \frac{\delta}{2} + \frac{1}{2n} - \frac{1}{n} \\ &= 1 - H_q(\tfrac{\delta}{4}) - \tfrac{\delta}{2}. \end{aligned}$$

□

Capítulo 3

Decodificação de Códigos de Pares de Símbolos

Nesse Capítulo apresentamos três algoritmos de decodificação para os códigos de pares de símbolos. O primeiro, apresentado por Cassuto e Blaum [3], que reduz o problema de decodificação de pares de erros a um problema de decodificação de erros e exclusões na métrica de Hamming. O segundo algoritmo usa o que definimos como *síndrome de pares* e *síndrome do símbolo vizinho* para decodificar códigos binários. Este decodificador foi proposto por Hirotomo, Takita e Morii [14]. O terceiro, proposto por Yaakobi, Bruck e Siegel em [28] e [29], decodifica códigos cíclicos binários.

Vimos que na teoria clássica de códigos corretores de erros, um decodificador é uma aplicação $\mathcal{D}_{\mathcal{C}} : \mathcal{A}^n \longrightarrow \mathcal{C} \cup \{F\}$, com F o conjunto das falhas do decodificador, no qual as entradas são vetores $y \in \mathcal{A}^n$. Na métrica de pares, um decodificador é uma aplicação $\mathcal{D}_{\pi(\mathcal{C})} : (\mathcal{A}, \mathcal{A})^n \longrightarrow \mathcal{C} \cup \{F\}$ que recebe vetores de pares de $(\mathcal{A}, \mathcal{A})^n$ e tem como resultado um vetor de $\mathcal{C}$ ou uma falha F, caso são seja possível corrigir tal vetor recebido.

3.1 Algoritmo de Decodificação de Cassuto-Blaum

Lembramos que se $x \in \mathcal{A}^n$, então $\pi(x) = [(x_0, x_1), (x_1, x_2), \ldots, (x_{n-1}, x_0)]$, isto é, a i-ésima coordenada x_i de x aparece em dois pares consecutivos de $\pi(x)$, a saber, (x_{i-1}, x_i) e (x_i, x_{i+1}).

Seja $u = [(u_{0,0}, u_{0,1}), (u_{1,0}, u_{1,1}), \ldots, (u_{n-1,0}, u_{n-1,1})]$ um vetor de pares recebido. Se, para algum i, $u_{i,0} \neq u_{i-1,1}$ (isto é, as entradas em pares consecutivos são diferentes) diremos que houve um **apagamento** na palavra recebida.

Definição 3.1.1. *Uma* **cadeia de l pares consecutivos de erros** *em u é uma sequência de pares $(u_{i,0}, u_{i,1}), (u_{i+1,0}, u_{i+1,1}), \ldots, (u_{i+l-1,0}, u_{i+l-1,1})$ todos com erro e tal que $u_{i,0}$ e $u_{i+l,1}$ estão corretos. Isto é, todos os pares $(u_{i+j,0}, u_{i+j,1})$, com $0 \leq j \leq l-1$, estão com erro e os pares $(u_{i-1,0}, u_{i-1,1})$ e $(u_{i+l,0}, u_{i+l,1})$ estão corretos.*

Suponha que um vetor de pares $u = [(u_{0,0}, u_{0,1}), (u_{1,0}, u_{1,1}), \ldots, (u_{n-1,0}, u_{n-1,1})]$ foi recebido.

Algoritmo de Decodificação 1:Defina um vetor $z = (z_0, z_1, \ldots, z_{n-1})$ com n símbolos do seguinte modo

$$z_i = \begin{cases} u_{i,0}, & \text{se } u_{i,0} = u_{i-1,1} \\ *, & \text{caso contrário.} \end{cases}$$

O símbolo * representa um apagamento. Para decodificar, basta aplicar um algoritmo de decodificação da métrica de Hamming no vetor z.

A questão a ser feita é se este algoritmo garante a Proposição 2.1.24, isto é, quando ele permite encontrar uma única palavra (se existir) dentro de uma bola de raio $[(d_P - 1)/2]$ ao redor da palavra recebida. A resposta é *não* em geral, e *sim* para códigos intercalados.

De fato, consideremos o código corretor de pares de erros $\{00000, 01100\}$ com 2 palavras e $d_P = 3$. Suponha que o vetor de pares $u = [(0,0), (1,1), (0,0), (0,0), (0,0)]$ seja recebido. Então o Algoritmo 1 vai transformar u em $z = (0, *, *, 0, 0)$ e o decodificador da métrica de Hamming vai falhar em decodificá-la (pois ambas as palavras do código são igualmente prováveis). Por outro lado, o decodificador de pares vai diferenciar que u está a uma distância de pares 1 de 00000 e a distância de pares 2 de 01100, escolhendo o vetor 00000 dentro da bola de raio 1.

Assim, transformar a decodificação de vetores de pares em uma decodificação na métrica de Hamming utilizando o Algoritmo 1 não é uma boa escolha, em geral. Entretanto este algoritmo é capaz de corrigir os códigos intercalados (Definição 2.2.1) dentro da capacidade de correção de um código de pares (Proposição 2.1.24) como é demostrado no próximo Teorema.

Teorema 3.1.2. *Seja $\mathcal{C}$ um $(2n, M_1 M_2, d_H)$-código construído instercalando dois códigos $\mathcal{C}_1$ e $\mathcal{C}_2$, cada um com distância de Hamming d_H. Então o decodificador do Algoritmo 1 pode corrigir até $[(d_P - 1)/2]$ pares de erros.*

Demonstração. Sejam $c = (x_0, y_0, x_1, y_1, \ldots, x_{n-1}, y_{n-1})$ o vetor enviado, no qual

$x = (x_0, x_1, \ldots, x_{n-1}) \in \mathcal{C}_1$ e $y = (y_0, y_1, \ldots, y_{n-1}) \in \mathcal{C}_2$ e

$$u = [(u_{0,0}, u_{0,1}), (u_{1,0}, u_{1,1}), \ldots, (u_{2n-1,0}, u_{2n-1,1})] \tag{3.1}$$

o vetor recebido com t pares de erros, $t \leq \lfloor \frac{d_p - 1}{2} \rfloor$ (uma versão com erros de $\pi(c)$). Observe que se

$$(u_{i,0}, u_{i,1}), (u_{i+1,0}, u_{i+1,1}), \ldots, (u_{i+l,0}, u_{i+l,1}) \tag{3.2}$$

é uma cadeia de l pares de erros consecutivos, para l ímpar, temos pelo menos uma entrada correta em cada código $\mathcal{C}_i$, $i = 1, 2$, e até $\frac{l-1}{2}$ entradas erradas em cada um destes códigos; para l par, um dos códigos terá 2 entradas corretas e até $\frac{l}{2} - 1$ entradas erradas e o outro não terá entradas corretas e terá até $\frac{l}{2}$ entradas erradas. Assim, para cada cadeia (3.2), cada código $\mathcal{C}_i$, $i = 1, 2$, tem a soma $2 \cdot \sharp$*número de entradas erradas* $+ \sharp$*número de entradas corretas* no máximo l. Desta maneira, para cada palavra do tipo (3.1), a soma dos números $2 \cdot \sharp$*número de entradas erradas* $+ \sharp$*número de entradas corretas*, percorrendo todas as possíveis cadeias do tipo (3.2) em (3.1), deve ser no máximo t (para cada código $\mathcal{C}_i$, $i = 1, 2$), já que t é o número de pares de erros do vetor u. Substituindo $d_P = 2d_H$ do Teorema 6 na Proposição 3, temos $t \leq \lfloor (d_P - 1)/2 \rfloor = \lfloor (2d_H - 1)/2 \rfloor = d_H - \lceil 1/2 \rceil = d_H - 1$ e, pela teoria clássica de códigos, o Algoritmo de decodificação do código de Hamming poderá corrigir erro/exclusões de soma $d_H - 1$. □

Exemplo 3.1.3. *Para exemplificar a demonstração do Teorema 3.1.2, suponha que* $c = (x_0, y_0, x_1, y_1, \ldots, x_{n-1}, y_{n-1})$ *seja o vetor armazenado, no qual* $x = (x_0, x_1, \ldots, x_{n-1}) \in \mathcal{C}_1$ *e* $y = (y_0, y_1, \ldots, y_{n-1}) \in \mathcal{C}_2$*,*

$$u = [(u_{0,0}, u_{0,1}), (u_{1,0}, u_{1,1}), \ldots, (u_{2n-1,0}, u_{2n-1,1})]$$

seja o vetor de pares recebido pelo decodificador e

$$(u_{i,0}, u_{i,1}), (u_{i+1,0}, u_{i+1,1}), (u_{i+2,0}, u_{i+2,1}), (u_{i+3,0}, u_{i+3,1}), (u_{i+4,0}, u_{i+4,1}) \tag{3.3}$$

$$(u_{j,0}, u_{j,1}), (u_{j+1,0}, u_{j+1,1}), (u_{j+2,0}, u_{j+2,1}), (u_{j+3,0}, u_{j+3,1}) \tag{3.4}$$

sejam duas cadeias de pares de erros de u*.*

Observe que a cadeia (3.3) *possui 5 pares de erros (*l *um número ímpar). Se* $u_{i,0}$ *representar uma coordenada de* $x \in \mathcal{C}_1$ *então* $u_{i+1,1}, u_{i+2,0}, u_{i+3,1}, u_{i+4,0}$ *também representam coordenadas do vetor* x *e* $u_{i,1}, u_{i+1,0}, u_{i+2,1}, u_{i+3,0}, u_{i+4,1}$ *são representações de coordenadas do vetor* $y \in \mathcal{C}_2$*. Como* $u_{i,0}$ *e* $u_{i+4,1}$ *são entradas corretas pela Definição 3.1.1, dizemos que ambos os códigos possuem uma coordenada correta. Além disso,* $\mathcal{C}_1$ *e* $\mathcal{C}_2$ *possuem até 2 entradas erradas cada (*$\frac{l-1}{2}$*), pois* $u_{i+1,1}$ *e* $u_{i+2,0}$ *são representações da mesma coordenada,*

o mesmo acontece para $u_{i+3,1}$ e $u_{i+4,0}$, $u_{i,1}$ e $u_{i+1,0}$ e, $u_{i+2,1}$ e $u_{i+3,0}$. O caso em que $u_{i,0}$ representa uma coordenada de $y \in \mathcal{C}_2$ é análogo.

Agora, a cadeia (3.4) *possui 4 pares de erro (l par). Se $u_{j,0}$ representar uma coordenada de $x \in \mathcal{C}_1$ então $u_{j+1,1}, u_{j+2,0}, u_{j+3,1}$ também representam coordenadas de x e $u_{j,1}, u_{j+1,0}, u_{j+2,1}, u_{j+3,0}$ representam coordenadas do vetor $y \in \mathcal{C}_2$. Como $u_{i,0}$ e $u_{i+4,1}$ são entradas corretas, $\mathcal{C}_1$ possui 2 coordenada corretas em u e até 1 coordenada com erro $(\frac{l}{2} - 1)$ pois $u_{j+2,0}$ e $u_{j+3,1}$ são representação da mesma coordenada em x. E o código $\mathcal{C}_2$ não possui coordenadas corretas e possui até 2 entradas com erro $(\frac{l}{2})$ pois $u_{j,1}$ e $u_{j+1,0}$ são representações da mesma coordenada de y, o mesmo acontece com $u_{j+2,1}$ e $u_{j+3,0}$. O caso em que $u_{j,0}$ representa uma coordenada de $y \in \mathcal{C}_2$, $\mathcal{C}_2$ terá 2 coordenadas corretas em u e até 1 erro e $\mathcal{C}_2$ não terá coordenadas corretas e terá até 2 coordenadas erradas.*

Exemplo 3.1.4. *Sejam $\mathcal{C}_1 = \{0000000, 1101010\}$ e $\mathcal{C}_2 = \{0000000, 1001111, 1111001, 0110110\}$ códigos sobre $\mathbb{F}_2$ com distância mínima de Hamming $d_H = 4$. Então o código obtido intercalando $\mathcal{C}_1$ e $\mathcal{C}_2$ é*

$$\begin{aligned}\mathcal{C} \;=\; &\{00000000000000, 01000001010101, 01010101000001, 00010100010100,\\ &10100010001000, 11100011011101, 11110111001001, 10110110011100\}\end{aligned}$$

com $d_H(\mathcal{C}) = 4$, $d_P(\mathcal{C}) = 8$ e capacidade de correção de até $t = \lfloor \frac{d_P(\mathcal{C})-1}{2} \rfloor = 3$.

Seja $c = (10100010001000)$ uma palavra armazenada e suponha que $y = [(10), (01), (01), (10), (00), (01), (10), (11), (00), (01), (10), (00), (00), (01)]$ seja um vetor de pares recebido, então de acordo com o Algoritmo 1, temos

$$z = (10 * 1001 * *01000).$$

Note que o vetor mais próximo de z é (10100010001000).

3.2 Algoritmo de decodificação por síndrome

Para exibir o segundo decodificador definimos primeiramente a matriz teste de paridade para pares de símbolos e a síndrome de pares.

Definição 3.2.1. *Seja $H = [h_0, h_1, \ldots, h_{n-k-1}]^T$ a matriz teste de paridade de um código linear em blocos, com h_i a i-ésima linha da matriz H. Representando cada linha da matriz teste de paridade pelo seu vetor de pares de símbolos, obtemos*

$$\pi(H) = \begin{bmatrix} \pi(h_0) \\ \pi(h_1) \\ \vdots \\ \pi(h_{n-k-1}) \end{bmatrix} = \begin{bmatrix} (h_{0,0}, h_{0,1}) & \cdots & (h_{0,n-1}, h_{0,0}) \\ (h_{1,0}, h_{1,1}) & \cdots & (h_{1,n-1}, h_{1,0}) \\ \vdots & & \vdots \\ (h_{n-k-1,0}, h_{n-k-1,1}) & \cdots & (h_{n-k-1,n-1}, h_{n-k-1,0}) \end{bmatrix}$$

A matriz $\pi(H)$ é chamada de **matriz teste de paridade de pares de símbolos**.

Seja u um vetor de pares recebido. Calculamos a síndrome multiplicando u pela transposta da matriz teste de paridade de pares de símbolos.

$$s^{(p)} = u \cdot \pi(H)^T.$$

Chamamos $s^{(p)}$ de **síndrome de pares** *do vetor u.*

Para realizar este cálculo precisamos do produto interno entre o vetor de pares e a linha da matriz teste de paridade de pares de símbolos.

Definição 3.2.2. *O produto interno entre dois vetores de pares*

$$u = [(u_{0,0}, u_{0,1}), (u_{1,0}, u_{1,1}), \ldots, (u_{n-1,0}, u_{n-1,1})] \quad e$$

$$v = [(v_{0,0}, v_{0,1}), (v_{1,0}, v_{1,1}), \ldots, (v_{n-1,0}, v_{n-1,1})]$$

é dado por

$$u \cdot v = [(u_{0,0} \cdot v_{0,0} + u_{1,0} \cdot v_{1,0} + \ldots + u_{n-1,0} \cdot v_{n-1,0}, u_{0,1} \cdot v_{0,1} + u_{1,1} \cdot v_{1,1} + \ldots + u_{n-1,1} \cdot v_{n-1,1})].$$

Por exemplo, o produto de $[(0,1),(0,0),(1,0)]$ e $[(0,0),(1,0),(1,0)]$ é dado por

$$[(0,1),(0,0),(1,0)] \cdot [(0,0),(1,0),(1,0)] = [(0 \cdot 0 + 0 \cdot 1 + 1 \cdot 1, 1 \cdot 0 + 0 \cdot 0 + 0 \cdot 0)] = [(1,0)].$$

No cálculo da síndrome de pares, os símbolos da direita e da esquerda são calculados independentemente da mesma forma que a síndrome na métrica de Hamming. Entretanto, diferentes pares de erros podem possuir a mesma síndrome de pares.

Exemplo 3.2.3. *Considere o $(3,2)$-código binário $\mathcal{C}_1$ com matriz teste de paridade dada por*

$$H = [1,1,1].$$

Note que $d_H(\mathcal{C}_1) = 1$, assim $\mathcal{C}$ pode detectar um erro e não pode corrigir nenhum erro.

Entretanto, seu código de pares é

$$\pi(\mathcal{C}_1) = \{[(0,0),(0,0),(0,0)],[(1,1),(1,0),(0,1)],[(0,1),(1,1),(1,0)],[(1,0),(0,1),(1,1)]\},$$

e $d_P(\mathcal{C}) = 3$*, isto é,* $\mathcal{C}_1$ *pode corrigir* 1 *par de erros. A matriz teste de paridade desse código é*

$$\pi(H) = [(1,1),(1,1),(1,1)].$$

Logo, a síndrome de pares dos três vetores

$$\begin{aligned} e_1 &= [(0,1),(0,0),(0,0)] \\ e_2 &= [(0,0),(0,1),(0,0)] \\ e_3 &= [(0,0),(0,0),(0,1)] \end{aligned}$$

é $[(0,1)]$.

Lema 3.2.4. *Sejam* $\mathcal{C}$ *um código linear com matriz teste de paridade* H *e* $y = [(c_{1,0},c_{2,1}),(c_{1,1},c_{2,2}),\ldots,(c_{1,n-1},c_{2,0})] \in (\mathcal{A},\mathcal{A})^n$ *um vetor de pares com* $c_1 = (c_{1,0},c_{1,1},\ldots,c_{1,n-1})$*,* $c_2 = (c_{2,0},c_{2,1},\ldots,c_{2,n-1}) \in \mathcal{A}^n$*. Então* $y \cdot \pi(H)^T = \mathbf{0}$ *se, e somente se,* $c_1, c_2 \in \mathcal{C}$*.*

Demonstração. Seja $\mathcal{C}$ um código com matriz teste de paridade

$$H = [h_0, h_1, \ldots, h_{n-k-1}]^T = \begin{bmatrix} h_{0,0} & h_{0,1} & \ldots & h_{0,n-1} \\ h_{1,0} & h_{1,1} & \ldots & h_{1,n-1} \\ \vdots & \vdots & & \vdots \\ h_{n-k-1,0} & h_{n-k-1,1} & \ldots & h_{n-k-1,n-1} \end{bmatrix}.$$

Assim, a matriz teste de paridade de pares de símbolos é

$$\pi(H) = \begin{bmatrix} (h_{0,0},h_{0,1}) & \cdots & (h_{0,n-1},h_{0,0}) \\ (h_{1,0},h_{1,1}) & \cdots & (h_{1,n-1},h_{1,0}) \\ \vdots & & \vdots \\ (h_{n-k-1,0},h_{n-k-1,1}) & \cdots & (h_{n-k-1,n-1},h_{n-k-1,0}) \end{bmatrix}$$

Sabemos que $c \in \mathcal{C}$ se, e somente se $Hc^T = \mathbf{0}$, isto é,

$$c \in \mathcal{C} \Leftrightarrow \begin{cases} h_{0,0}c_0 & +\ldots+ & h_{0,n-1}c_{n-1} & = 0 \\ \vdots & & & \\ h_{n-k-1,0}c_0 & +\ldots+ & h_{n-k-1,n-1}c_{n-1} & = 0. \end{cases} \tag{3.5}$$

Seja $y = [(c_{1,0}, c_{2,1}), (c_{1,1}, c_{2,2}), \ldots, (c_{1,n-1}, c_{2,0})] \in (\mathcal{A}, \mathcal{A})^n$ um vetor de pares com $c_1 = (c_{1,0}, c_{1,1}, \ldots, c_{1,n-1}), c_2 = (c_{2,0}, c_{2,1}, \ldots, c_{2,n-1}) \in \mathcal{A}^n$. Então $y \cdot \pi(H)^T = \mathbf{0}$ se, e somente se,

$s^{(p)} = y \cdot \pi(H)^T = [(c_{1,0}h_{0,0} + \ldots + c_{1,n-1}h_{0,n-1}, c_{2,0}h_{0,1} + \ldots + c_{2,n-1}h_{0,0}), \ldots, (c_{1,0}h_{n-k-1,0} + \ldots + c_{1,n-1}h_{n-k-1,n-1}, c_{2,0}h_{n-k-1,1} + \ldots + c_{2,n-1}h_{n-k-1,0})] = [(0,0), \ldots, (0,0)]$, isto é, se, e somente se

$$\begin{cases} h_{0,0}c_{1,0} & + \ldots + & h_{0,n-1}c_{1,n-1} & = 0 \\ \vdots & & & \\ h_{n-k-1,0}c_{1,0} & + \ldots + & h_{n-k-1,n-1}c_{1,n-1} & = 0 \end{cases}$$

$$\begin{cases} h_{0,0}c_{2,0} & + \ldots + & h_{0,n-1}c_{2,n-1} & = 0 \\ \vdots & & & \\ h_{n-k-1,0}c_{2,0} & + \ldots + & h_{n-k-1,n-1}c_{2,n-1} & = 0. \end{cases}$$

Por (3.5), isto acontece se, e somente se, $c_1, c_2 \in \mathcal{C}$.

□

Corolário 3.2.5. *Seja $\mathcal{C}$ um código linear com matriz teste de paridade H. Então $\pi(c) \cdot \pi(H)^T = \mathbf{0}$, para todo $c \in \mathcal{C}$.*

Demonstração. Utilizando as notações do Lema 3.2.4, se $c = (c_0, c_1, \ldots, c_{n-1}) \in \mathcal{C}$ temos $\pi(c) \cdot \pi(H)^T = [(c_0h_{0,0} + \ldots + c_{n-1}h_{0,n-1}, c_1h_{0,1} + \ldots + c_0h_{0,0}), \ldots, (c_0h_{n-k-1,0} + \ldots + c_{n-1}h_{n-k-1,n-1}, c_1h_{n-k-1,1} + \ldots + c_0h_{n-k-1,0})]$. Assim, por (3.5),

$$\pi(c) \cdot \pi(H)^T = [(0,0), \ldots, (0,0)].$$

□

Como a síndrome de pares não é única para cada vetor erro (Exemplo 3.2.3), isto não nos permite determinar um algoritmo preciso para corrigir pares de erros dentro da capacidade de correção de um código.

Precisamos definir uma nova síndrome que indica qual par de um vetor recebido tem erro e, para isto, usaremos a definição de disjunção exclusiva de símbolos. Apesar desta definição valer para qualquer alfabeto, nem todas as propriedades que ela satisfaz para o alfabeto binário são válidas, em geral para alfabeto não binário.

A partir de agora consideramos $\mathcal{A} = \{0, 1\}$.

Definição 3.2.6. *Sejam $a, b \in \mathcal{A}$, então*

$$a \oplus^1 b = \begin{cases} 0, & se\ a = b \\ 1, & se\ a \neq b \end{cases}$$

Por exemplo, $2 \oplus 1 = 1$ e $2 \oplus 2 = 0$.

p	q	$p \oplus q$
0	0	0
0	1	1
1	0	1
1	1	0

Tabela 3.1: Tabela verdade.

Pela Tabela 3.1, podemos verificar as seguintes propriedades que serão úteis na demonstração do Teorema 3.2.11

- (Comutatividade) $a \oplus b = b \oplus a$, para todos $a, b \in \mathcal{A}$.
- (Associatividade) $a \oplus (b \oplus c) = (a \oplus b) \oplus c$, para todos $a, b, c \in \mathcal{A}$.
- (Elemento neutro) $0 \oplus a = a$, para todo $a \in \mathcal{A}$.

De forma análoga, definimos a disjunção exclusiva entre dois vetores de pares.

Definição 3.2.7. *Sejam $u, v \in (\mathcal{A}, \mathcal{A})^n$ com*

$$u = [(u_{0,0}, u_{0,1}), (u_{1,0}, u_{1,1}), \ldots, (u_{n-1,0}, u_{n-1,1})]$$

e

$$v = [(v_{0,0}, v_{0,1}), (v_{1,0}, v_{1,1}), \ldots, (v_{n-1,0}, v_{n-1,1})].$$

Defina

$$u \oplus v = [(u_{0,0} \oplus v_{0,0}, u_{0,1} \oplus v_{0,1}), (u_{1,0} \oplus v_{1,0}, u_{1,1} \oplus v_{1,1}), \ldots, (u_{n-1,0} \oplus v_{n-1,0}, u_{n-1,1} \oplus v_{n-1,1}))].$$

De forma análoga a definição para disjunção exclusiva para símbolos, a disjunção exclusiva entre vetores de pares preservam as mesmas propriedades.

Para todos $u, v, w \in (\mathcal{A}, \mathcal{A})^n$, temos

[1] Este símbolo, comumente usado em Matemática para denotar soma direta, aqui é utilizado para denotar a disjunção exclusiva denotada comumente na lógica por $\veebar$.

- (Comutatividade) $u \oplus v = v \oplus u$, para todos $u, v \in (\mathcal{A}, A)^n$.
- (Associatividade) $u \oplus (v \oplus w) = (u \oplus v) \oplus w$, para todos $u, v, w \in (\mathcal{A}, \mathcal{A})^n$.
- (Elemento neutro) $\mathbf{0} \oplus u = u$, para todo $u \in (\mathcal{A}, \mathcal{A})^n$.

Exemplo 3.2.8. *Sejam* $u = [(1,0),(1,1),(0,0),(0,1)]$ *e* $v = [(1,0),(0,1),(1,0),(1,1)]$ *vetores de pares de* $(\mathcal{A}, \mathcal{A})^4$. *Então,*

$$u \oplus v = [(1 \oplus 1, 0 \oplus 0), (1 \oplus 0, 1 \oplus 1), (0 \oplus 1, 0 \oplus 0), (0 \oplus 1, 1 \oplus 1)] = [(0,0),(1,0),(1,0),(1,0)].$$

Com a Definição 3.2.6, podemos definir uma nova síndrome para que seja possível corrigir pares de erros dentro da capacidade de correção de um código.

Definição 3.2.9. *Dado um vetor de pares*

$$u = [(u_{0,0}, u_{0,1}), (u_{1,0}, u_{1,1}), \ldots, (u_{n-1,0}, u_{n-1,1})],$$

definimos sua **síndrome do símbolo vizinho** $s^{(n)} = (s_0^{(n)}, s_1^{(n)}, \ldots, s_{n-1}^{(n)})$, *tal que*

$$s_i^{(n)} = \begin{cases} 0, & \text{se } u_{i,0} = u_{i-1,1} \\ 1, & \text{se } u_{i,0} \neq u_{i-1,1}. \end{cases}$$

Se o símbolo da esquerda do i-ésimo par e o símbolo da direita do $(i-1)$-ésimo par são iguais, então o i-ésimo símbolo da síndrome do símbolo vizinho é 0, caso contrário, o i-ésimo símbolo da síndrome do símbolo vizinho é 1.

O i-ésimo termo da síndrome do símbolo vizinho também pode ser obtido pela disjunção exclusiva

$$s_i^{(n)} = u_{i,0} \oplus u_{i-1,1},$$

isto é, a síndrome do símbolo vizinho de u pode ser obtida por

$$s^{(n)} = (u_{0,0} \oplus u_{n-1,1}, u_{1,0} \oplus u_{0,1}, \ldots, u_{n-1,0} \oplus u_{n-2,1}).$$

Exemplo 3.2.10. *Se o vetor* $[(0,0),(1,1),(1,0),(0,0),(0,0)]$ *é recebido, a síndrome do símbolo vizinho é* $s^{(n)} = (01000)$. *Isso significa que o símbolo da esquerda do segundo par e o símbolo da direita do primeiro par do vetor recebido são diferentes.*

No teorema a seguir determinamos condições necessárias para garantir que os vetores de pares de classes distintas possuam síndromes distintas.

Teorema 3.2.11. *Se um código* $\mathcal{C}$ *pode corrigir* t *pares de erros, o par* $(s^{(p)}, s^{(n)})$ *é único, para cada vetor erro* e *com* $\omega_P(e) \leq t$.

Demonstração. Sejam e e e' dois vetores de pares distintos com $\omega_P(e), \omega_P(e') \leq t$. A síndrome de pares e a síndrome do símbolo vizinho de e e e' são dados por

$$\begin{aligned} s^{(p)} &= e \cdot \pi(H)^T, \\ s^{(n)} &= (e_{0,0} \oplus e_{n-1,1}, e_{1,0} \oplus e_{0,1}, \ldots, e_{n-1,0} \oplus e_{n-2,1}), \\ s^{(p)'} &= e' \cdot \pi(H)^T, \\ s^{(n)'} &= (e'_{0,0} \oplus e'_{n-1,1}, e'_{1,0} \oplus e'_{0,1}, \ldots, e'_{n-1,0} \oplus e'_{n-2,1}). \end{aligned}$$

Suponha que a síndrome de pares e a síndrome do símbolo vizinho de e e e' sejam as mesmas, isto é, $s^{(p)} = s^{(p)'}$ e $s^{(n)} = s^{(n)'}$. Logo

$$e \cdot \pi(H)^T = e' \cdot \pi(H)^T \Leftrightarrow (e - e') \cdot \pi(H)^T = \mathbf{0}.$$

Note que

$$\begin{cases} e_{i,j} - e'_{i,j} = 0 & \Leftrightarrow \quad e'_{i,j} = e_{i,j} \\ e_{i,j} - e'_{i,j} = 1 & \Leftrightarrow \quad e'_{i,j} \neq e_{i,j} \end{cases}$$

Assim, $e - e' = e \oplus e'$, logo

$$s^{(p)} = s^{(p')} \Leftrightarrow (e \oplus e') \cdot \pi(H)^T = \mathbf{0}.$$

Daí, pelo Lema 3.2.4, existem $c_1 = (c_{1,0}, c_{1,1}, \ldots, c_{1,n-1})$ e $c_2 = (c_{2,0}, c_{2,1}, \ldots, c_{2,n-1})$ palavras de $\mathcal{C}$ tais que

$$\begin{aligned} e \oplus e' &= [(c_{1,0}, c_{2,1}), (c_{1,1}c_{2,2}), \ldots, (c_{1,n-1}, c_{2,0})] \Rightarrow \\ e \oplus (e \oplus e') &= e \oplus [(c_{1,0}, c_{2,1}), (c_{1,1}c_{2,2}), \ldots, (c_{1,n-1}, c_{2,0})] \Rightarrow \\ (e \oplus e) \oplus e' &= e \oplus [(c_{1,0}, c_{2,1}), (c_{1,1}c_{2,2}), \ldots, (c_{1,n-1}, c_{2,0})] \Rightarrow \\ 0 \oplus e' &= e \oplus [(c_{1,0}, c_{2,1}), (c_{1,1}c_{2,2}), \ldots, (c_{1,n-1}, c_{2,0})] \Rightarrow \\ e' &= e \oplus [(c_{1,0}, c_{2,1}), (c_{1,1}c_{2,2}), \ldots, (c_{1,n-1}, c_{2,0})] \end{aligned} \tag{3.6}$$

assim

$$e' = [(e_{0,0} \oplus c_{1,0}, e_{0,1} \oplus c_{2,1}), (e_{1,0} \oplus c_{1,1}, e_{1,1} \oplus c_{2,2}), \ldots, (e_{n-1,0} \oplus c_{1,n-1}, e_{n-1,1} \oplus c_{2,0})]. \tag{3.7}$$

Além disso, como a síndrome do símbolo vizinho de e e e' são iguais, os pares com erro de e são correspondentes aos de e'. Daí:

i) se $e_{i,0} = e_{i-1,1}$, temos $e'_{i,0} = e'_{i-1,1}$. Segue de (3.7)

$$e'_{i,0} = e_{i,0} \oplus c_{1,i} \text{ e } e'_{i-1,1} = e_{i-1,1} \oplus c_{2,i},$$

assim $e_{i,0} \oplus c_{1,i} = e_{i-1,1} \oplus c_{2,i}$ o que implica

$e_{i,0} \oplus e_{i,0} \oplus c_{1,i} = e_{i,0} \oplus e_{i-1,1} \oplus c_{2,i} = e_{i-1,1} \oplus e_{i-1,1} \oplus c_{2,i}$. Logo

$0 \oplus c_{1,i} = 0 \oplus c_{2,i} \Rightarrow c_{1,i} = c_{2,i}$.

ii) se $e_{j,0} \neq e_{j-1,1}$, temos $e'_{j,0} \neq e'_{j-1,1}$ e, como $s_j^{(n)} = s_j^{(n)'}$, segue

$$e_{j,0} \oplus e_{j-1,1} = 1 = e'_{j,0} \oplus e'_{j-1,1}. \tag{3.8}$$

De (3.7), temos $e'_{j,0} = e_{j,0} \oplus c_{1,j}$ e $e'_{j-1,1} = e_{j-1,1} \oplus c_{2,j}$, assim

$$\begin{aligned} e'_{j-1,1} \oplus e'_{j,0} &= e'_{j-1,1} \oplus e_{j,0} \oplus c_{1,j} \overset{(3.8)}{\Rightarrow} \\ e_{j,0} \oplus e_{j-1,1} &= e_{j,0} \oplus e'_{j-1,1} \oplus c_{1,j} = e_{j,0} \oplus (e_{j-1,1} \oplus c_{2,j}) \oplus c_{1,j} \end{aligned}$$

Da associatividade da operação e por (3.8)

$$\begin{aligned} 1 &= 1 \oplus c_{2,j} \oplus c_{1,j} \Rightarrow \\ 1 \oplus c_{1,j} &= 1 \oplus c_{2,j} \oplus c_{1,j} \oplus c_{1,j} = 1 \oplus c_{2,j} \Rightarrow \\ 1 \oplus 1 \oplus c_{1,j} &= 1 \oplus 1 \oplus c_{2,j} \Rightarrow \\ 0 \oplus c_{1,j} &= 0 \oplus c_{2,j} \Rightarrow \\ c_{1,j} &= c_{2,j} \end{aligned}$$

Logo, c_1 e c_2 são as mesmas palavras não nulas. Portanto, de (3.7), os vetores erro satisfazem

$$e' = e \oplus [(c_{1,0}, c_{1,1}), (c_{1,1}, c_{1,2}), \ldots, (c_{1,n-1}, c_{1,0})] = e \oplus \pi(c_1).$$

Como a distância mínima de pares de $\mathcal{C}$ é $d_P(\mathcal{C}) = 2t + 1$, então

$$\begin{aligned} \omega_P(c_1) &= \omega_P(e' - e) = d_P(e', e) \\ &\leq d_P(e', \mathbf{0}) + d_P(e, \mathbf{0}) \\ &= \omega_P(e') + \omega_P(e) \\ \omega_P(e') &\geq \omega_P(c_1) - \omega_P(e) \\ &\geq 2t + 1 - t \\ &= t + 1, \end{aligned}$$

contradizendo a hipótese. Logo, o par $(s^{(p)}, s^{(n)})$ é único para cada vetor erro e com $\omega_P(e) \leq t$. □

Seja $y \in (\mathcal{A}, \mathcal{A})^n$. Defina

$$y + \pi(\mathcal{C}) = \{y + \pi(c);\ c \in \mathcal{C}\}.$$

Proposição 3.2.12. *Seja $\mathcal{C}$ um (n, k)-código linear. Temos*

(i) $y + \pi(\mathcal{C}) = y' + \pi(\mathcal{C}) \Leftrightarrow y - y' \in \pi(\mathcal{C})$;

(ii) $(y + \pi(\mathcal{C})) \cap (y' + \pi(\mathcal{C})) \neq \emptyset \Rightarrow y + \pi(\mathcal{C}) = y' + \pi(\mathcal{C})$;

(iii) $\bigcup_{y \in (\mathcal{A},\mathcal{A})^n} (y + \pi(\mathcal{C})) = (\mathcal{A}, \mathcal{A})^n$;

(iv) $\mid (y + \pi(\mathcal{C})) \mid = \mid \mathcal{C} \mid = q^k$.

Demonstração. (i) $y + \pi(\mathcal{C}) = y' + \pi(\mathcal{C}) \Leftrightarrow$ existem $c, c' \in \mathcal{C}$ tais que $y + \pi(c) = y' + \pi(c') \Leftrightarrow y - y' = \pi(c') - \pi(c) \Leftrightarrow y - y' \in \pi(\mathcal{C})$.

(ii) Se $(y + \pi(\mathcal{C})) \cap (y' + \pi(\mathcal{C})) \neq \emptyset$, então existe $x \in (y + \pi(\mathcal{C})) \cap (y' + \pi(\mathcal{C}))$, ou seja, $x \in y + \pi(\mathcal{C})$ e $x \in y' + \pi(\mathcal{C})$. Daí, existem $c, c' \in \mathcal{C}$ tais que $x = y + \pi(c)$ e $x = y' + \pi(c')$, assim

$$\begin{aligned} y + \pi(c) = y' + \pi(c') &\Rightarrow y - y' = \pi(c') - \pi(c) \\ &\Rightarrow y - y' \in \pi(\mathcal{C}) \\ &\overset{\text{(i)}}{\Rightarrow} y + \pi(\mathcal{C}) = y' + \pi(\mathcal{C}). \end{aligned}$$

(iii) Dado $y \in (\mathcal{A}, \mathcal{A})^n$, $y \in y + \pi(\mathcal{C})$, pois $0 \in \mathcal{C}$ (já que $\mathcal{C}$ é espaço vetorial) e $\pi(0) = 0$ em $(\mathcal{A}, \mathcal{A})^n$. Logo $(\mathcal{A}, \mathcal{A})^n \subset \bigcup_{y \in (\mathcal{A},\mathcal{A})^n} (y + \pi(\mathcal{C}))$. Além disso, $\bigcup_{y \in (\mathcal{A},\mathcal{A})^n} (y + \pi(\mathcal{C})) \subset (\mathcal{A}, \mathcal{A})^n$, pois $y + \pi(\mathcal{C}) \subset (\mathcal{A}, \mathcal{A})^n$, para todo $y \in (\mathcal{A}, \mathcal{A})^n$. Portanto,

$$\bigcup_{y \in (\mathcal{A},\mathcal{A})^n} (y + \pi(\mathcal{C})) = (\mathcal{A}, \mathcal{A})^n.$$

(iv) A função $\begin{array}{rcl} \varphi : & \mathcal{C} & \longrightarrow y + \pi(\mathcal{C}) \\ & c & \longmapsto y + \pi(c) \end{array}$ é uma bijeção. Sejam $c, c' \in \mathcal{C}$ tais que

$$\varphi(c) = \varphi(c') \Rightarrow y + \pi(c) = y + \pi(c') \Rightarrow \pi(c) = \pi(c') \Rightarrow c = c',$$

então φ é injetiva. Além disso, dado $x \in y + \pi(\mathcal{C})$, existe $c \in \mathcal{C}$ tal que $x = y + \pi(c) = \varphi(c)$, logo φ é sobrejetiva. Portanto, como φ é uma bijeção,

$$\mid y + \pi(\mathcal{C}) \mid = \mid \mathcal{C} \mid = q^k.$$

□

Cada conjunto da forma $y + \pi(\mathcal{C}) = \{y + \pi(c);\ c \in \mathcal{C}\}$ é chamado de **classe lateral**

de y segundo $\pi(\mathcal{C})$. Note o seguinte

$$y + \pi(\mathcal{C}) = \pi(\mathcal{C}) \Longleftrightarrow y \in \pi(\mathcal{C}).$$

De (ii)-(iv), temos que o número de classes laterais segundo $\pi(\mathcal{C})$ é

$$\frac{q^{2n}}{q^k} = q^{2n-k}.$$

Exemplo 3.2.13. *Seja $\mathcal{C}$ o $(3,3)$-código linear gerado sobre $\mathbb{F}_2$ pela matriz*

$$G = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}.$$

Assim, $\mathcal{C} = \{000, 100, 010, 001, 110, 101, 011, 111\}$,

$$\begin{aligned} \pi(\mathcal{C}) \;=\; & \{[(00),(00),(00)],[(10),(00),(01)],[(01),(10),(00)],[(00),(01),(10)], \\ & [(11),(10),(01)],[(10),(01),(11)],[(01),(11),(10)],[(11),(11),(11)]\} \end{aligned}$$

e as classes laterais segundo $\pi(\mathcal{C})$ são

$$\begin{aligned} [(00),(00),(00)] + \pi(\mathcal{C}) \;=\; & \{[(00),(00),(00)],[(10),(00),(01)],[(01),(10),(00)], \\ & [(00),(01),(10)],[(11),(10),(01)],[(10),(01),(11)], \\ & [(01),(11),(10)],[(11),(11),(11)]\} \end{aligned}$$

$$\begin{aligned} [(10),(00),(00)] + \pi(\mathcal{C}) \;=\; & \{[(10),(00),(00)],[(00),(00),(01)],[(11),(10),(00)], \\ & [(10),(01),(10)],[(01),(10),(01)],[(00),(01),(11)], \\ & [(11),(11),(10)],[(01),(11),(11)]\} \end{aligned}$$

$$\begin{aligned} [(01),(00),(00)] + \pi(\mathcal{C}) \;=\; & \{[(01),(00),(00)],[(11),(00),(01)],[(00),(10),(00)], \\ & [(01),(01),(10)],[(10),(10),(01)],[(11),(01),(11)], \\ & [(00),(11),(10)],[(10),(11),(11)]\} \end{aligned}$$

$$\begin{aligned} [(11),(00),(00)] + \pi(\mathcal{C}) \;=\; & \{[(11),(00),(00)],[(01),(00),(01)],[(10),(10),(00)], \\ & [(11),(01),(10)],[(00),(10),(01)],[(01),(01),(11)], \\ & [(10),(11),(10)],[(00),(11),(11)]\} \end{aligned}$$

$$\begin{aligned}[(00),(10),(00)]+\pi(\mathcal{C}) &= \{[(00),(10),(00)],[(10),(10),(01)],[(01),(00),(00)],\\ &\quad [(00),(11),(10)],[(11),(00),(01)],[(10),(11),(11)],\\ &\quad [(01),(01),(10)],[(11),(01),(11)]\}\end{aligned}$$

$$\begin{aligned}[(00),(01),(00)]+\pi(\mathcal{C}) &= \{[(00),(01),(00)],[(10),(01),(01)],[(01),(11),(00)],\\ &\quad [(00),(00),(10)],[(11),(11),(01)],[(10),(00),(11)],\\ &\quad [(01),(10),(10)],[(11),(10),(11)]\}\end{aligned}$$

$$\begin{aligned}[(00),(11),(00)]+\pi(\mathcal{C}) &= \{[(00),(11),(00)],[(10),(11),(01)],[(01),(01),(00)],\\ &\quad [(00),(10),(10)],[(11),(01),(01)],[(10),(10),(11)],\\ &\quad [(01),(00),(10)],[(11),(00),(11)]\}\end{aligned}$$

$$\begin{aligned}[(00),(00),(11)]+\pi(\mathcal{C}) &= \{[(00),(00),(11)],[(10),(00),(10)],[(01),(10),(11)],\\ &\quad [(00),(01),(01)],[(11),(10),(10)],[(10),(01),(00)],\\ &\quad [(01),(11),(01)],[(11),(11),(00)]\}\end{aligned}$$

Corolário 3.2.14. *Sejam $u, v \in (\mathcal{A},\mathcal{A})^n$ tais que $u \in v + \pi(\mathcal{C})$, então o par $(s^{(p)}, s^{(n)})$, das síndromes de pares e da síndrome do símbolo vizinho, de u e v são iguais.*

Demonstração. Se $u \in v + \pi(\mathcal{C})$, então existe $c \in \mathcal{C}$ tal que $u = v + \pi(c)$. Daí, $u \cdot \pi(H)^T = (v + \pi(c)) \cdot \pi(H)^T = v \cdot \pi(H)^T + \pi(c) \cdot \pi(H)^T$, assim pelo Corolário 3.2.5, $u \cdot \pi(H)^T = v \cdot \pi(H)^T$, isto é, as síndromes de pares de u e v são iguais.

Agora, sejam $c = (c_0, c_1, \ldots, c_{n-1})$, $u = [(u_{0,0}, u_{0,1}), (u_{1,0}, u_{1,1}), \ldots, (u_{n-1,0}, u_{n-1,1})]$ e $v = [(v_{0,0}, v_{0,1}), (v_{1,0}, v_{1,1}), \ldots, (v_{n-1,0}, v_{n-1,1})]$, então

$$u = v + \pi(c) = [(v_{0,0} + c_0, v_{0,1} + c_1), (v_{1,0} + c_1, v_{1,1} + c_2), \ldots, (v_{n-1,0} + c_{n-1}, v_{n-1,1} + c_0)].$$

Daí,

i) Se $u_{i,0} = u_{i-1,1}$, temos $v_{i,0} + c_i = v_{i-1,1} + c_i$, ou seja, $v_{i,0} = v_{i-1,1}$.

ii) Se $u_{i,0} \neq u_{i-1,1}$, então $v_{i,0} + c_i \neq v_{i-1,1} + c_i$, isto é, $v_{i,0} \neq v_{i-1,1}$.

Logo, a síndrome do símbolo vizinho $s^{(n)}$ de u e v são iguais.

□

Do Corolário 3.2.14 temos que todos os elementos de uma mesma classe lateral têm o mesmo par de síndromes $(s^{(p)}, s^{(n)})$.

Definição 3.2.15. *Um vetor de pares de peso mínimo numa classe lateral é chamado de* **elemento líder** *dessa classe.*

No Exemplo 3.2.13 temos

- $[(00),(00),(00)]$ é elemento líder de $\pi(\mathcal{C})$,
- $[(10),(00),(00)]$ é líder de $[(10),(00),(00)]+\pi(\mathcal{C})$,
- $[(01),(00),(00)]$ é líder de $[(01),(00),(00)]+\pi(\mathcal{C})$,
- $[(11),(00),(00)]$ é líder de $[(11),(00),(00)]+\pi(\mathcal{C})$,
- $[(00),(10),(00)]$ é líder de $[(00),(10),(00)]+\pi(\mathcal{C})$,
- $[(00),(01),(00)]$ é líder de $[(00),(01),(00)]+\pi(\mathcal{C})$,
- $[(00),(11),(00)]$ é líder de $[(00),(11),(00)]+\pi(\mathcal{C})$,
- $[(00),(00),(11)]$ é líder de $[(00),(00),(11)]+\pi(\mathcal{C})$.

Proposição 3.2.16. *Seja $\mathcal{C}$ um código linear em $\mathcal{A}^n$ com distância mínima de pares $d_P = d_P(\mathcal{C})$. Se $u \in (\mathcal{A},\mathcal{A})^n$ é tal que*

$$\omega_H(u) \leq t_P,$$

com $t_P = \left\lfloor \frac{d_P-1}{2} \right\rfloor$, então u é o único elemento líder de sua classe.

Demonstração. Suponhamos que existam $u, v \in y+\pi(\mathcal{C})$ tais que

$$\omega_H(u), \omega_H(v) \leq t_P,$$

então

$$\omega_H(u-v) = d_H(u,v) \leq d_H(u,0)+d_H(v,0) = \omega_H(u)+\omega_H(v) \leq 2\cdot\left\lfloor \frac{d_P-1}{2} \right\rfloor \leq d_P-1.$$

Além disso, como $u, v \in y+\pi(\mathcal{C})$, existem $c, c' \in \mathcal{C}$ tais que $u = y+\pi(c)$ e $v = y+\pi(c')$. Daí,

$$\omega_H(u-v) = \omega_H(y+\pi(c)-(y+\pi(c')) = \omega_H(\pi(c)-\pi(c')) = \omega_H(\pi(c-c')) = \omega_P(c-c') \geq d_P,$$

o que é um absurdo. Logo $u = v$. □

Apresentamos agora o algoritmo de decodificação por síndromes para códigos de pares de símbolos lineares binários. Como preparação para esse algoritmo, determine todos os elementos y de $(\mathcal{A}, \mathcal{A})^n$ tais que $\omega_H(y) \leq t_P$. Em seguida calcule a síndrome de pares e a síndrome do símbolo vizinho desses elementos e coloque esses dados em uma tabela. Seja

$$u = [(u_{0,0}, u_{0,1}), (u_{1,0}, u_{1,1}), \ldots, (u_{n-1,0}, u_{n-1,1})]$$

um vetor recebido.

Algoritmo de Decodificação 2:

Passo 1. Calcule a síndrome de pares e a síndrome do símbolo vizinho do vetor de pares u.

$$\begin{aligned} s^{(p)} &= u \cdot \pi(H)^T \\ s^{(n)} &= (u_{0,0} \oplus u_{n-1,1}, u_{1,0} \oplus u_{0,1}, \ldots, u_{n-1,0} \oplus u_{n-2,1}). \end{aligned}$$

Passo 2. Localize o líder e da classe de elementos cujas síndrome de pares e síndrome do símbolo vizinho são iguais à $s^{(p)}$ e $s^{(n)}$, respectivamente.

Passo 3. Assuma que e é o vetor erro e decodifique o vetor de pares

$$\pi(w) = u \oplus e.$$

Passo 4. Transforme o vetor de pares $\pi(w)$ na palavra

$$w = (w_0, w_1, \ldots, w_{n-1}).$$

■

Lembre que, para códigos de pares de símbolos com distância miníma de pares $d_P(\mathcal{C}) = 2t + 1$, este algoritmo pode corrigir até t pares de erros, com $t \leq \lfloor (d_P(\mathcal{C}) - 1)/2 \rfloor$, como na Proposição 2.1.24.

Exemplo 3.2.17. *Considere o* $(3, 2)$*-código do Exemplo* (3.2.3)*, como* $d_P(\mathcal{C}) = 3$*, o Algoritmo 2 pode corrigir até* $t_P = \left\lfloor \frac{d_P(\mathcal{C})-1}{2} \right\rfloor = 1$.

Os vetores de pares com peso ≤ 1 *e suas respectivas síndromes de pares e síndromes do símbolo vizinho estão relacionados na tabela a seguir*

líder	*síndrome de pares* ($s^{(p)}$)	*síndrome do símbolo vizinho* ($s^{(n)}$)
[(00),(00),(00)]	*[(00)]*	*(000)*
[(10),(00),(00)]	*[(10)]*	*(100)*
[(01),(00),(00)]	*[(01)]*	*(010)*
[(11),(00),(00)]	*[(11)]*	*(110)*
[(00),(10),(00)]	*[(10)]*	*(010)*
[(00),(01),(00)]	*[(01)]*	*(001)*
[(00),(11),(00)]	*[(11)]*	*(011)*
[(00),(00),(10)]	*[(10)]*	*(001)*
[(00),(00),(01)]	*[(01)]*	*(100)*
[(00),(00),(11)]	*[(11)]*	*(101)*

Suponha que o vetor de pares recebido seja $u = [(01), (10), (10)]$.

Passo 1: $s^{(p)} = u \cdot \pi(H)^T = [(01)]$ *e* $s^{(n)} = (001)$

Passo 2: $e = [(00), (01), (00)]$

Passo 3: $\pi(c) = u \oplus e = [(01), (10), (10)] \oplus [(00), (01), (00)] = [(01), (11), (10)]$

Passo 4: $c_1 = (011)$.

Exemplo 3.2.18. *Seja* $\mathcal{C}$ *o código cíclico linear gerado por* $g(x) = 1 + x + x^3$. *A matriz teste de paridade de* $\mathcal{C}$ *é dada por*

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

A distância mínima de Hamming de $\mathcal{C}$ *é 3, então* $\mathcal{C}$ *pode corrigir* $t_H = 1$ *erros de símbolos. Além disso,* $d_P(\mathcal{C}) = 5$ *e o Algoritmo 2 pode corrigir até* $t_P = \left\lfloor \frac{d_P(\mathcal{C})-1}{2} \right\rfloor = 2$ *pares de de erros. Sua matriz teste de paridade de pares de símbolos é*

$$\pi(H) = \begin{pmatrix} (1,0) & (0,0) & (0,1) & (1,0) & (0,1) & (1,1) & (1,1) \\ (0,1) & (1,0) & (0,1) & (1,1) & (1,1) & (1,0) & (0,0) \\ (0,0) & (0,1) & (1,0) & (0,1) & (1,1) & (1,1) & (1,0) \end{pmatrix}.$$

Os vetores de pares com peso ≤ 2 *e suas respectivas síndromes de pares e síndromes do símbolo vizinho estão relacionados na tabela abaixo*

líder	$s^{(p)}$	$s^{(n)}$
[(00),(00),(00),(00),(00),(00),(00)]	*[(00),(00),(00)]*	*(0000000)*
[(10),(00),(00),(00),(00),(00),(00)]	*[(10),(00),(00)]*	*(1000000)*
[(01),(00),(00),(00),(00),(00),(00)]	*[(00),(01),(00)]*	*(0100000)*
[(11),(00),(00),(00),(00),(00),(00)]	*[(10),(01),(00)]*	*(1100000)*
[(00),(10),(00),(00),(00),(00),(00)]	*[(00),(10),(00)]*	*(0100000)*
[(00),(01),(00),(00),(00),(00),(00)]	*[(00),(00),(01)]*	*(0010000)*
[(00),(11),(00),(00),(00),(00),(00)]	*[(00),(10),(01)]*	*(0110000)*
[(00),(00),(10),(00),(00),(00),(00)]	*[(00),(00),(10)]*	*(0010000)*
[(00),(00),(01),(00),(00),(00),(00)]	*[(01),(01),(00)]*	*(0001000)*
[(00),(00),(11),(00),(00),(00),(00)]	*[(01),(01),(10)]*	*(0011000)*
[(00),(00),(00),(10),(00),(00),(00)]	*[(10),(10),(00)]*	*(0001000)*
[(00),(00),(00),(01),(00),(00),(00)]	*[(00),(01),(01)]*	*(0000100)*
[(00),(00),(00),(11),(00),(00),(00)]	*[(10),(11),(01)]*	*(0001100)*
[(00),(00),(00),(00),(10),(00),(00)]	*[(00),(10),(10)]*	*(0000100)*
[(00),(00),(00),(00),(01),(00),(00)]	*[(01),(01),(01)]*	*(0000010)*
[(00),(00),(00),(00),(11),(00),(00)]	*[(01),(11),(11)]*	*(0000110)*
[(00),(00),(00),(00),(00),(10),(00)]	*[(10),(10),(10)]*	*(0000010)*
[(00),(00),(00),(00),(00),(01),(00)]	*[(01),(00),(01)]*	*(0000001)*
[(00),(00),(00),(00),(00),(11),(00)]	*[(11),(10),(11)]*	*(0000011)*
[(00),(00),(00),(00),(00),(00),(10)]	*[(10),(00),(10)]*	*(0000001)*
[(00),(00),(00),(00),(00),(00),(01)]	*[(01),(00),(00)]*	*(1000000)*
[(00),(00),(00),(00),(00),(00),(11)]	*[(11),(00),(10)]*	*(1000001)*
[(10),(10),(00),(00),(00),(00),(00)]	*[(10),(10),(00)]*	*(1100000)*
[(10),(01),(00),(00),(00),(00),(00)]	*[(10),(00),(01)]*	*(1010000)*
[(10),(11),(00),(00),(00),(00),(00)]	*[(10),(10),(01)]*	*(1110000)*
[(01),(10),(00),(00),(00),(00),(00)]	*[(00),(11),(00)]*	*(0000000)*
[(01),(01),(00),(00),(00),(00),(00)]	*[(00),(01),(01)]*	*(0110000)*
[(01),(11),(00),(00),(00),(00),(00)]	*[(00),(11),(01)]*	*(0010000)*
[(11),(10),(00),(00),(00),(00),(00)]	*[(10),(11),(00)]*	*(1000000)*
[(11),(01),(00),(00),(00),(00),(00)]	*[(10),(01),(01)]*	*(1110000)*
[(11),(11),(00),(00),(00),(00),(00)]	*[(10),(11),(01)]*	*(1010000)*
[(10),(00),(10),(00),(00),(00),(00)]	*[(10),(00),(10)]*	*(1010000)*
[(10),(00),(01),(00),(00),(00),(00)]	*[(11),(01),(00)]*	*(1001000)*
[(10),(00),(11),(00),(00),(00),(00)]	*[(11),(01),(10)]*	*(1011000)*
[(01),(00),(10),(00),(00),(00),(00)]	*[(00),(01),(10)]*	*(0110000)*
[(01),(00),(01),(00),(00),(00),(00)]	*[(01),(00),(00)]*	*(0101000)*
[(01),(00),(11),(00),(00),(00),(00)]	*[(01),(00),(10)]*	*(0111000)*
[(11),(00),(10),(00),(00),(00),(00)]	*[(10),(01),(10)]*	*(1110000)*
[(11),(00),(01),(00),(00),(00),(00)]	*[(11),(00),(00)]*	*(1101000)*
[(11),(00),(11),(00),(00),(00),(00)]	*[(11),(00),(10)]*	*(1111000)*
[(10),(00),(00),(10),(00),(00),(00)]	*[(00),(10),(00)]*	*(1001000)*
[(10),(00),(00),(01),(00),(00),(00)]	*[(10),(01),(01)]*	*(1000100)*
[(10),(00),(00),(11),(00),(00),(00)]	*[(00),(11),(01)]*	*(1001100)*
[(01),(00),(00),(10),(00),(00),(00)]	*[(10),(11),(00)]*	*(0101000)*
[(01),(00),(00),(01),(00),(00),(00)]	*[(00),(00),(01)]*	*(0100100)*
[(01),(00),(00),(11),(00),(00),(00)]	*[(10),(10),(01)]*	*(0101100)*
[(11),(00),(00),(10),(00),(00),(00)]	*[(00),(11),(00)]*	*(1101000)*
[(11),(00),(00),(01),(00),(00),(00)]	*[(10),(00),(01)]*	*(1100100)*
[(11),(00),(00),(11),(00),(00),(00)]	*[(00),(10),(01)]*	*(1101100)*
[(10),(00),(00),(00),(10),(00),(00)]	*[(10),(10),(10)]*	*(1000100)*
[(10),(00),(00),(00),(01),(00),(00)]	*[(11),(01),(01)]*	*(1000010)*

líder	$s^{(p)}$	$s^{(n)}$
[(10),(00),(00),(00),(11),(00),(00)]	*[(11),(11),(11)]*	*(1000110)*
[(01),(00),(00),(00),(10),(00),(00)]	*[(00),(11),(10)]*	*(0100100)*
[(01),(00),(00),(00),(01),(00),(00)]	*[(01),(00),(01)]*	*(0100010)*
[(01),(00),(00),(00),(11),(00),(00)]	*[(01),(10),(11)]*	*(0100110)*
[(11),(00),(00),(00),(10),(00),(00)]	*[(10),(11),(10)]*	*(1100100)*
[(11),(00),(00),(00),(01),(00),(00)]	*[(11),(00),(01)]*	*(1100010)*
[(11),(00),(00),(00),(11),(00),(00)]	*[(11),(10),(11)]*	*(1100110)*
[(10),(00),(00),(00),(00),(10),(00)]	*[(00),(10),(10)]*	*(1000010)*
[(10),(00),(00),(00),(00),(01),(00)]	*[(11),(00),(01)]*	*(1000001)*
[(10),(00),(00),(00),(00),(11),(00)]	*[(01),(10),(11)]*	*(1000011)*
[(01),(00),(00),(00),(00),(10),(00)]	*[(10),(11),(10)]*	*(0100010)*
[(01),(00),(00),(00),(00),(01),(00)]	*[(01),(01),(01)]*	*(0100001)*
[(01),(00),(00),(00),(00),(11),(00)]	*[(11),(11),(11)]*	*(0100011)*
[(11),(00),(00),(00),(00),(10),(00)]	*[(00),(11),(10)]*	*(1100010)*
[(11),(00),(00),(00),(00),(01),(00)]	*[(11),(01),(01)]*	*(1100001)*
[(11),(00),(00),(00),(00),(11),(00)]	*[(01),(11),(11)]*	*(1100011)*
[(10),(00),(00),(00),(00),(00),(10)]	*[(00),(00),(10)]*	*(1000001)*
[(10),(00),(00),(00),(00),(00),(01)]	*[(11),(00),(00)]*	*(0000000)*
[(10),(00),(00),(00),(00),(00),(11)]	*[(01),(00),(10)]*	*(0000001)*
[(01),(00),(00),(00),(00),(00),(10)]	*[(10),(01),(10)]*	*(0100001)*
[(01),(00),(00),(00),(00),(00),(01)]	*[(01),(01),(00)]*	*(1100000)*
[(01),(00),(00),(00),(00),(00),(11)]	*[(11),(01),(10)]*	*(1100001)*
[(11),(00),(00),(00),(00),(00),(10)]	*[(00),(01),(10)]*	*(1100001)*
[(11),(00),(00),(00),(00),(00),(01)]	*[(11),(01),(00)]*	*(0100000)*
[(11),(00),(00),(00),(00),(00),(11)]	*[(01),(01),(10)]*	*(0100001)*
[(00),(10),(10),(00),(00),(00),(00)]	*[(00),(10),(10)]*	*(0110000)*
[(00),(10),(01),(00),(00),(00),(00)]	*[(01),(11),(00)]*	*(0101000)*
[(00),(10),(11),(00),(00),(00),(00)]	*[(01),(11),(10)]*	*(0111000)*
[(00),(01),(10),(00),(00),(00),(00)]	*[(00),(00),(11)]*	*(0000000)*
[(00),(01),(01),(00),(00),(00),(00)]	*[(01),(01),(01)]*	*(0011000)*
[(00),(01),(11),(00),(00),(00),(00)]	*[(01),(01),(11)]*	*(0001000)*
[(00),(11),(10),(00),(00),(00),(00)]	*[(00),(10),(11)]*	*(0100000)*
[(00),(11),(01),(00),(00),(00),(00)]	*[(01),(11),(01)]*	*(0111000)*
[(00),(11),(11),(00),(00),(00),(00)]	*[(01),(11),(11)]*	*(0101000)*
[(00),(10),(00),(10),(00),(00),(00)]	*[(10),(00),(00)]*	*(0101000)*
[(00),(10),(00),(01),(00),(00),(00)]	*[(00),(11),(01)]*	*(0100100)*
[(00),(10),(00),(11),(00),(00),(00)]	*[(10),(01),(01)]*	*(0101100)*
[(00),(01),(00),(10),(00),(00),(00)]	*[(10),(10),(01)]*	*(0011000)*
[(00),(01),(00),(01),(00),(00),(00)]	*[(00),(01),(00)]*	*(0010100)*
[(00),(01),(00),(11),(00),(00),(00)]	*[(10),(11),(00)]*	*(0011100)*
[(00),(11),(00),(10),(00),(00),(00)]	*[(10),(00),(01)]*	*(0111000)*
[(00),(11),(00),(01),(00),(00),(00)]	*[(00),(11),(00)]*	*(0110100)*
[(00),(11),(00),(11),(00),(00),(00)]	*[(10),(01),(00)]*	*(0111100)*
[(00),(10),(00),(00),(10),(00),(00)]	*[(00),(00),(10)]*	*(0100100)*
[(00),(10),(00),(00),(01),(00),(00)]	*[(01),(11),(01)]*	*(0100010)*
[(00),(10),(00),(00),(11),(00),(00)]	*[(01),(01),(11)]*	*(0100110)*
[(00),(01),(00),(00),(10),(00),(00)]	*[(00),(10),(11)]*	*(0010100)*
[(00),(01),(00),(00),(01),(00),(00)]	*[(01),(01),(00)]*	*(0010010)*
[(00),(01),(00),(00),(11),(00),(00)]	*[(01),(11),(10)]*	*(0010110)*
[(00),(11),(00),(00),(10),(00),(00)]	*[(00),(00),(11)]*	*(0110100)*
[(00),(11),(00),(00),(01),(00),(00)]	*[(01),(11),(00)]*	*(0110010)*

líder	$s^{(p)}$	$s^{(n)}$
[(00),(11),(00),(00),(11),(00),(00)]	*[(01),(01),(10)]*	*(0110110)*
[(00),(10),(00),(00),(00),(10),(00)]	*[(10),(00),(10)]*	*(0100010)*
[(00),(10),(00),(00),(00),(01),(00)]	*[(01),(10),(01)]*	*(0100001)*
[(00),(10),(00),(00),(00),(11),(00)]	*[(11),(00),(11)]*	*(0100011)*
[(00),(01),(00),(00),(00),(10),(00)]	*[(10),(10),(11)]*	*(0010010)*
[(00),(01),(00),(00),(00),(01),(00)]	*[(01),(00),(00)]*	*(0010001)*
[(00),(01),(00),(00),(00),(11),(00)]	*[(11),(10),(10)]*	*(0010011)*
[(00),(11),(00),(00),(00),(10),(00)]	*[(10),(00),(11)]*	*(0110010)*
[(00),(11),(00),(00),(00),(01),(00)]	*[(01),(10),(00)]*	*(0110001)*
[(00),(11),(00),(00),(00),(11),(00)]	*[(11),(00),(10)]*	*(0110011)*
[(00),(10),(00),(00),(00),(00),(10)]	*[(10),(10),(10)]*	*(0100001)*
[(00),(10),(00),(00),(00),(00),(01)]	*[(01),(10),(00)]*	*(1100000)*
[(00),(10),(00),(00),(00),(00),(11)]	*[(11),(10),(10)]*	*(1100001)*
[(00),(01),(00),(00),(00),(00),(10)]	*[(10),(00),(11)]*	*(0010001)*
[(00),(01),(00),(00),(00),(00),(01)]	*[(01),(00),(01)]*	*(1010000)*
[(00),(01),(00),(00),(00),(00),(11)]	*[(11),(00),(11)]*	*(1010001)*
[(00),(11),(00),(00),(00),(00),(10)]	*[(10),(10),(11)]*	*(0110001)*
[(00),(11),(00),(00),(00),(00),(01)]	*[(01),(10),(01)]*	*(1110000)*
[(00),(11),(00),(00),(00),(00),(11)]	*[(11),(10),(11)]*	*(1110001)*
[(00),(00),(10),(10),(00),(00),(00)]	*[(10),(10),(10)]*	*(0011000)*
[(00),(00),(10),(01),(00),(00),(00)]	*[(00),(01),(11)]*	*(0010100)*
[(00),(00),(10),(11),(00),(00),(00)]	*[(10),(11),(11)]*	*(0011100)*
[(00),(00),(01),(10),(00),(00),(00)]	*[(11),(11),(00)]*	*(0000000)*
[(00),(00),(01),(01),(00),(00),(00)]	*[(01),(00),(01)]*	*(0001100)*
[(00),(00),(01),(11),(00),(00),(00)]	*[(11),(10),(01)]*	*(0000100)*
[(00),(00),(11),(10),(00),(00),(00)]	*[(11),(11),(10)]*	*(0010000)*
[(00),(00),(11),(01),(00),(00),(00)]	*[(01),(00),(11)]*	*(0011100)*
[(00),(00),(11),(11),(00),(00),(00)]	*[(11),(10),(11)]*	*(0010100)*
[(00),(00),(10),(00),(10),(00),(00)]	*[(00),(10),(00)]*	*(0010100)*
[(00),(00),(10),(00),(01),(00),(00)]	*[(01),(01),(11)]*	*(0010010)*
[(00),(00),(10),(00),(11),(00),(00)]	*[(01),(11),(01)]*	*(0010110)*
[(00),(00),(01),(00),(10),(00),(00)]	*[(01),(11),(10)]*	*(0001100)*
[(00),(00),(01),(00),(01),(00),(00)]	*[(00),(00),(01)]*	*(0001010)*
[(00),(00),(01),(00),(11),(00),(00)]	*[(00),(10),(11)]*	*(0001110)*
[(00),(00),(11),(00),(10),(00),(00)]	*[(01),(11),(00)]*	*(0011100)*
[(00),(00),(11),(00),(01),(00),(00)]	*[(00),(00),(11)]*	*(0011010)*
[(00),(00),(11),(00),(11),(00),(00)]	*[(00),(10),(01)]*	*(0011110)*
[(00),(00),(10),(00),(00),(10),(00)]	*[(10),(10),(00)]*	*(0010010)*
[(00),(00),(10),(00),(00),(01),(00)]	*[(01),(00),(11)]*	*(0010001)*
[(00),(00),(10),(00),(00),(11),(00)]	*[(11),(10),(01)]*	*(0010011)*
[(00),(00),(01),(00),(00),(10),(00)]	*[(11),(11),(10)]*	*(0001010)*
[(00),(00),(01),(00),(00),(01),(00)]	*[(00),(01),(01)]*	*(0001001)*
[(00),(00),(01),(00),(00),(11),(00)]	*[(10),(11),(11)]*	*(0001011)*
[(00),(00),(11),(00),(00),(10),(00)]	*[(11),(11),(00)]*	*(0011010)*
[(00),(00),(11),(00),(00),(01),(00)]	*[(00),(01),(11)]*	*(0011001)*
[(00),(00),(11),(00),(00),(11),(00)]	*[(10),(11),(01)]*	*(0011011)*
[(00),(00),(10),(00),(00),(00),(10)]	*[(10),(00),(00)]*	*(0010001)*
[(00),(00),(10),(00),(00),(00),(01)]	*[(01),(00),(10)]*	*(1010000)*
[(00),(00),(10),(00),(00),(00),(11)]	*[(11),(00),(00)]*	*(1010001)*
[(00),(00),(01),(00),(00),(00),(10)]	*[(11),(01),(10)]*	*(0001001)*
[(00),(00),(01),(00),(00),(00),(01)]	*[(00),(01),(00)]*	*(1001000)*
[(00),(00),(01),(00),(00),(00),(11)]	*[(10),(01),(10)]*	*(1001001)*
[(00),(00),(11),(00),(00),(00),(10)]	*[(11),(01),(00)]*	*(0011001)*
[(00),(00),(11),(00),(00),(00),(01)]	*[(00),(01),(10)]*	*(1011000)*
[(00),(00),(11),(00),(00),(00),(11)]	*[(10),(01),(10)]*	*(1011001)*

líder	$s^{(p)}$	$s^{(n)}$
[(00),(00),(00),(10),(10),(00),(00)]	*[(10),(00),(10)]*	*(0001100)*
[(00),(00),(00),(10),(01),(00),(00)]	*[(11),(11),(01)]*	*(0001010)*
[(00),(00),(00),(10),(11),(00),(00)]	*[(11),(01),(11)]*	*(0001110)*
[(00),(00),(00),(01),(10),(00),(00)]	*[(00),(11),(11)]*	*(0000000)*
[(00),(00),(00),(01),(01),(00),(00)]	*[(01),(00),(00)]*	*(0000110)*
[(00),(00),(00),(01),(11),(00),(00)]	*[(01),(10),(10)]*	*(0000010)*
[(00),(00),(00),(11),(10),(00),(00)]	*[(10),(01),(11)]*	*(0001000)*
[(00),(00),(00),(11),(01),(00),(00)]	*[(11),(10),(00)]*	*(0001110)*
[(00),(00),(00),(11),(11),(00),(00)]	*[(11),(00),(10)]*	*(0001010)*
[(00),(00),(00),(10),(00),(10),(00)]	*[(00),(00),(10)]*	*(0001010)*
[(00),(00),(00),(10),(00),(01),(00)]	*[(11),(10),(01)]*	*(0001001)*
[(00),(00),(00),(10),(00),(11),(00)]	*[(01),(00),(11)]*	*(0001011)*
[(00),(00),(00),(01),(00),(10),(00)]	*[(10),(11),(11)]*	*(0000110)*
[(00),(00),(00),(01),(00),(01),(00)]	*[(01),(01),(00)]*	*(0000101)*
[(00),(00),(00),(01),(00),(11),(00)]	*[(11),(11),(10)]*	*(0000111)*
[(00),(00),(00),(11),(00),(10),(00)]	*[(00),(01),(11)]*	*(0001110)*
[(00),(00),(00),(11),(00),(01),(00)]	*[(11),(11),(00)]*	*(0001101)*
[(00),(00),(00),(11),(00),(11),(00)]	*[(01),(01),(10)]*	*(0001111)*
[(00),(00),(00),(10),(00),(00),(10)]	*[(00),(10),(10)]*	*(0001001)*
[(00),(00),(00),(10),(00),(00),(01)]	*[(11),(10),(00)]*	*(1001000)*
[(00),(00),(00),(10),(00),(00),(11)]	*[(01),(10),(10)]*	*(1001001)*
[(00),(00),(00),(01),(00),(00),(10)]	*[(10),(01),(11)]*	*(0000101)*
[(00),(00),(00),(01),(00),(00),(01)]	*[(01),(01),(01)]*	*(1000100)*
[(00),(00),(00),(01),(00),(00),(11)]	*[(11),(01),(11)]*	*(1000101)*
[(00),(00),(00),(11),(00),(00),(10)]	*[(00),(11),(11)]*	*(0001101)*
[(00),(00),(00),(11),(00),(00),(01)]	*[(11),(11),(01)]*	*(1001100)*
[(00),(00),(00),(11),(00),(00),(11)]	*[(01),(11),(11)]*	*(1001101)*
[(00),(00),(00),(00),(10),(10),(00)]	*[(10),(00),(00)]*	*(0000110)*
[(00),(00),(00),(00),(10),(01),(00)]	*[(01),(10),(11)]*	*(0000101)*
[(00),(00),(00),(00),(10),(11),(00)]	*[(11),(00),(01)]*	*(0000111)*
[(00),(00),(00),(00),(01),(10),(00)]	*[(11),(11),(11)]*	*(0000000)*
[(00),(00),(00),(00),(01),(01),(00)]	*[(00),(01),(00)]*	*(0000011)*
[(00),(00),(00),(00),(01),(11),(00)]	*[(10),(11),(10)]*	*(0000001)*
[(00),(00),(00),(00),(11),(10),(00)]	*[(11),(01),(01)]*	*(0000100)*
[(00),(00),(00),(00),(11),(01),(00)]	*[(00),(11),(10)]*	*(0000111)*
[(00),(00),(00),(00),(11),(11),(00)]	*[(10),(01),(00)]*	*(0000101)*
[(00),(00),(00),(00),(10),(00),(10)]	*[(10),(10),(00)]*	*(0000101)*
[(00),(00),(00),(00),(10),(00),(01)]	*[(01),(10),(10)]*	*(1000100)*
[(00),(00),(00),(00),(10),(00),(11)]	*[(11),(10),(00)]*	*(1000101)*
[(00),(00),(00),(00),(01),(00),(10)]	*[(11),(01),(11)]*	*(0000011)*
[(00),(00),(00),(00),(01),(00),(01)]	*[(00),(01),(01)]*	*(1000010)*
[(00),(00),(00),(00),(01),(00),(11)]	*[(10),(01),(11)]*	*(1000011)*
[(00),(00),(00),(00),(11),(00),(10)]	*[(11),(11),(01)]*	*(0000111)*
[(00),(00),(00),(00),(11),(00),(01)]	*[(00),(11),(11)]*	*(1000110)*
[(00),(00),(00),(00),(11),(00),(11)]	*[(10),(11),(01)]*	*(1000111)*
[(00),(00),(00),(00),(00),(10),(10)]	*[(00),(10),(00)]*	*(0000011)*
[(00),(00),(00),(00),(00),(10),(01)]	*[(11),(10),(10)]*	*(1000010)*
[(00),(00),(00),(00),(00),(10),(11)]	*[(01),(10),(00)]*	*(1000011)*
[(00),(00),(00),(00),(00),(01),(10)]	*[(11),(00),(11)]*	*(0000000)*
[(00),(00),(00),(00),(00),(01),(01)]	*[(00),(00),(01)]*	*(1000001)*
[(00),(00),(00),(00),(00),(01),(11)]	*[(10),(00),(11)]*	*(1000000)*
[(00),(00),(00),(00),(00),(11),(10)]	*[(01),(10),(01)]*	*(0000010)*
[(00),(00),(00),(00),(00),(11),(01)]	*[(10),(10),(11)]*	*(1000011)*
[(00),(00),(00),(00),(00),(11),(11)]	*[(00),(10),(01)]*	*(1000010)*

Suponha que a palavra recebida seja

1. $y_1 = [(0,1),(1,0),(0,1),(1,0),(0,0),(0,0),(0,1)]$

   Passo 1: $s_1^{(p)} = [(10),(00),(00)]$ *e* $s_1^{(n)} = (1000000)$

   Passo 2: $e_1 = [(10),(00),(00),(00),(00),(00),(00)]$

   Passo 3: $\pi(c_1) = y_1 \oplus e_1 = [(11),(10),(01),(10),(00),(00),(01)]$

   Passo 4: $c_1(1101000)$.

2. $y_2 = [(0,0),(1,1),(0,1),(1,1),(1,1),(0,1),(1,0)]$

   Passo 1: $s_2^{(p)} = [(00),(10),(01)]$ *e* $s_2^{(n)} = (0110000)$

   Passo 2: $e_2 = [(00),(11),(00),(00),(00),(00),(00)]$

   Passo 3: $\pi(c_2) = y_2 \oplus e_2 = [(00),(00),(01),(11),(10),(01),(10)]$

   Passo 4: $c_2 = (0001101)$.

3. $y_3 = [(0,0),(0,1),(0,1),(1,1),(1,0),(0,0),(1,0)]$

   Passo 1: $s_3^{(p)} = [(01),(00),(00)]$ *e* $s_3^{(n)} = (0010001)$

   Passo 2: $e_3 = [(00),(01),(00),(00),(00),(01),(00)]$

   Passo 3: $\pi(c_3) = y_3 \oplus e_3 = [(00),(00),(01),(11),(10),(01),(10)]$

   Passo 4: $c_3 = (0001101)$.

Para códigos de pares lineares com distância mínima de pares d_P, o algoritmo proposto nesta seção pode corrigir quaisquer t_P-pares de erros com $t_P \leq \left\lfloor \frac{d_P - 1}{2} \right\rfloor$, que é justamente a capacidade de correção de um código de pares como visto na Proposição 2.1.24.

3.3 Algoritmo de decodificação de Yaakobi-Bruck-Siegel

Pelo Teorema 2.4.3, concluímos que os códigos de pares construídos a partir de códigos cíclicos lineares binários têm uma distância mínima de pares maior, permitindo assim a correção de um número maior de erros. É, portanto, interessante construir decodificadores eficientes para esses códigos, o que é o tópico desta seção.

Observe que, como esses códigos são lineares, foi mostrado na Seção 3.2 como uma versão modificada da decodificação por síndrome pode ser usada para corrigir erros de pares de símbolos dentro da capacidade de correção de erros dos códigos. No entanto, a *decodificação por síndrome tem complexidade exponencial*, isto é, o número de síndromes

de pares de símbolos e de síndromes do símbolo vizinho necessárias cresce exponencialmente com relação ao comprimento e a distância mínima de pares do código. O objetivo dessa seção, de acordo com [28], é fornecer decodificadores mais eficientes, cuja complexidade seja da mesma ordem que os decodificadores clássicos para códigos cíclicos, visto que os códigos cíclicos são construídos sobre um anel de polinômios, o que torna a codificação e decodificação mais rápida e eficiente.

Seja $\mathcal{C}$ um código linear cíclico com distância mínima $d_H(\mathcal{C}) = 2t + 1$. Suponha que haja um decodificador para $\mathcal{C}$ que possa corrigir até t erros. Vamos mostrar como usar esse decodificador no projeto de construir um decodificador para o código $\pi(\mathcal{C})$ que corrige até $t_0 = \lfloor \frac{3t+1}{2} \rfloor$ erros de pares de símbolos, ou seja, complexidade exponencial.

Assuma que a dimensão de $\mathcal{C}$ seja maior que um. Essa condição implica, de acordo com a prova do Teorema 2.4.3, $d_H(\mathcal{C}) \leq \lfloor 2n/3 \rfloor$, ou seja, $2t + 1 \leq \lfloor 2n/3 \rfloor$. Daí

$$t \leq \lfloor \frac{\lfloor 2n/3 \rfloor - 1}{2} \rfloor < n/3.$$

Observe que $t_0 < n/2$. De fato,

$$t_0 = \lfloor \frac{3t+1}{2} \rfloor \leq \frac{3t+1}{2} \leq \frac{1}{2} \cdot \lfloor \frac{2n}{3} \rfloor + \frac{t}{2} < \frac{1}{2} \cdot \frac{2n}{3} + \frac{1}{2} \cdot \frac{n}{3} = \frac{n}{2}.$$

Definimos o decodificador como uma aplicação $\mathcal{D}_{\mathcal{C}} : \mathcal{A}^n \longrightarrow \mathcal{C} \cup \{F\}$, com F denotando as falhas do decodificador. Para uma palavra $y \in \mathcal{A}^n$, escrevemos $\mathcal{D}_{\mathcal{C}}(y) = \hat{c} \in \mathcal{C} \cup \{F\}$. Se $c \in \mathcal{C}$ é a palavra transmitida e $d_H(c, y) \leq t$, então $\hat{c} = c$ e dizemos que o decodificador foi **bem sucedido**. Entretanto, se $d_H(c, y) > t$, então ou $\hat{c}$ é uma palavra do código diferente de c, cuja distância de Hamming da palavra recebida y é no máximo t, isto é, $d_H(\hat{c}, y) \leq t$, ou $\hat{c} = F$, indicando que tal palavra não existe. Neste caso, dizemos que o decodificador **falhou** ou foi **mal sucedido**.

Agora introduzimos outro código que será usado no "algoritmo" de decodificação de pares de símbolos. O *código de dupla repetição* de $\mathcal{C}$ é o código definido por

$$\mathcal{C}_2 = \{(c, c);\ c \in \mathcal{C}\}.$$

Note que seu comprimento é $2n$ e sua distância de Hamming satisfaz $d_H(\mathcal{C}_2) = 2d_H(\mathcal{C})$. O código $\mathcal{C}_2$ pode corrigir até $2t$ erros e assumimos que ele tem um decodificador dado pela composição das funções f e g, a seguir.

$$\mathcal{D}_{\mathcal{C}_2} :\ \mathcal{A}^n \times \mathcal{A}^n \xrightarrow{f} \mathcal{C}_2 \cup \{F\} \xrightarrow{g} \mathcal{C} \cup \{F\}. \tag{3.9}$$

Quando é recebida uma palavra (x, y), então f retorna $(\hat{c}, \hat{c}) \in \mathcal{C}_2$ ou $\{F\}$. Se c foi a palavra transmitida e $d_H((c, c), (x, y))$ é no máximo $2t$, então $g(\hat{c}, \hat{c}) = \hat{c} = c \in \mathcal{C}$, isto é, $\mathcal{D}_{\mathcal{C}_2}(x, y) = c$. Agora, se $d_H((c, c), (x, y)) > t$, então ou $d_H((\hat{c}, \hat{c}), (x, y)) \leq t$, isto é, $(\hat{c}, \hat{c})$ é uma palavra de $\mathcal{C}_2$ diferente da enviada, ou $\mathcal{D}_{\mathcal{C}_2}(x, y) = g(F) = F$.

Considere uma palavra do código $c \in \mathcal{C}$ e seja $\pi(c) \in \pi(\mathcal{C})$ seu vetor de pares de símbolos correspondente. Seja $y \in \pi(c) + e$ a palavra recebida, com $e \in (\mathcal{A} \times \mathcal{A})^n$ o vetor erro com peso $\omega_P(e) \leq t_0 = \lfloor \frac{3t+1}{2} \rfloor$. Vamos descrever um decodificador

$$\mathcal{D}_\pi \ : \ (\mathcal{A}, \mathcal{A})^n \ \longrightarrow \ \{0, 1\}^n$$

que pode corrigir o erro e.

O vetor recebido tem a forma

$$y = ((y_{0,0}, y_{0,1}), (y_{1,0}, y_{1,1}), \ldots, (y_{n-1,0}, y_{n-1,1})).$$

Definimos três vetores relacionados:

$$\begin{aligned} y_L &= (y_{0,0}, \ldots, y_{n-1,0}), \\ y_R &= (y_{0,1}, \ldots, y_{n-1,1}), \\ y_S &= y_L + y_R \\ &= (y_{0,0} + y_{0,1}, \ldots, y_{n-1,0} + y_{n-1,1}). \end{aligned}$$

Como o vetor y sofre no máximo t_0 pares de erros, cada um dos vetores y_L e y_R tem no máximo t_0 erros. Pode-se pensar em y_L como uma versão com erros de $c = (c_0, c_1, \ldots, c_{n-1})$ e y_R como a versão com erros de um deslocamento cíclico à esquerda de c, $(c_1, \ldots, c_{n-1}, c_0)$. Além disso, y_L e o deslocamento cíclico à direita de y_R, $y_R^{(1)} = (y_{n-1,1}, y_{0,1}, \ldots, y_{n-2,1})$ podem ser vistos como duas versões com erros da mesma palavra c. Além disso, o vetor y_S tem no máximo t_0 erros com respeito à palavra código $c' = (c_0 + c_1, \ldots, c_{n-1} + c_0)$. No geral, a palavra c' não determina unicamente o valor de c. Entretanto, vamos mostrar agora que, nesta configuração, isto acontece. Este resultado, que usaremos no desenvolvimento do algoritmo de decodificação $\mathcal{D}_\pi$, é demonstrado no seguinte lema.

Lema 3.3.1. *Na configuração do canal de leitura de pares de símbolos descrita acima, se a palavra do código $c' \in \mathcal{C}$ é recuperada com sucesso, então podemos determinar unicamente a palavra c do código.*

Demonstração. Como a palavra $c' \in \mathcal{C}$ é decodificada com sucesso, sabemos o valor de

$$c' = (c'_0, c'_1, \ldots, c'_{n-1}).$$

Como $c' = (c_0 + c_1, c_1 + c_2, \ldots, c_{n-1} + c_0)$ e o alfabeto é $\{0, 1\}$, temos

$$\begin{cases} c'_0 &= c_0 + c_1 &\Rightarrow\ c_1 = c_0 + c'_0 \\ c'_1 &= c_1 + c_2 &\Rightarrow\ c_2 = c_0 + c'_0 + c'_1 \\ &\vdots & \\ c'_{n-2} &= c_{n-2} + c_{n-1} &\Rightarrow\ c_{n-1} = c_0 + c'_0 + c'_1 + \ldots + c'_{n-2}, \end{cases}$$

ou seja, a palavra c satisfaz $c_i = c_0 + \sum_{j=0}^{i-1} c'_j$, com $i = 0, \ldots, n-1$. Assim, se definirmos $\tilde{c} = (\tilde{c}_0, \ldots, \tilde{c}_{n-1})$, com $\tilde{c}_0 = 0$ e $\tilde{c}_i = \sum_{j=0}^{i-1} c'_j$, para $1 \leq i \leq n-1$, então a palavra do código c é igual à $\tilde{c}$ ou $\tilde{c} + \mathbf{1}$, dependendo do valor de c_0. A distância entre y_L e c é no máximo $t_0 = \lfloor \frac{3t+1}{2} \rfloor$ e $d_H(c, c + \mathbf{1}) = n$, pois estamos em $\mathbb{Z}_2$.

Se $(y_L)_i \neq c_i$ então ou $(y_L)_i = 1$ e $c_i = 0$ ($c_i + 1 = 1$), ou $(y_L)_i = 0$ e $c_i = 1$ ($c_i + 1 = 0$). Logo $(y_L)_i = c_i + 1$, ou seja, o número de coordenadas diferentes entre y_L e c é igual ao número de coordenadas iguais entre y_L e $c + \mathbf{1}$. Recordando que $t_0 < n/2$, temos

$$d_H(y_L, c + \mathbf{1}) = n - d_H(y_L, c) > n - t_0 > n - n/2 = n/2 > t_0. \tag{3.10}$$

Logo, se $d_H(y_L, \tilde{c}) < d_H(y_L, \tilde{c} + \mathbf{1})$, então $c = \tilde{c}$, pois se $c = \tilde{c} + \mathbf{1}$, teríamos $d_H(y_L, c + \mathbf{1}) < d_H(y_L, c) < t_0$, contradizendo (3.10). Caso contrário, se $d_H(y_L, \tilde{c}) \geq d_H(y_L, \tilde{c} + \mathbf{1})$, então $c = \tilde{c} + \mathbf{1}$, pois se $c = \tilde{c}$ então teríamos $d_H(y_L, c) \geq d_H(y_L, c + \mathbf{1}) > t_0$, contradizendo a hipótese. Em ambos os casos, podemos recuperar a palavra do código c. □

Por conveniência, denotamos a palavra do código c obtida da palavra c' pelo método do Lema 3.3.1 como c'^*, isto é, $c'^* = c$.

O número de erros de pares no vetor y é no máximo t_0. Cada erro num vetor de pares de símbolos corresponde a uma ou duas entradas erradas em um dos pares de símbolos. Seja E_1 o número de erros de pares de símbolos em apenas uma das entradas e E_2 o número de pares de erros de duas entradas, com $E_1 + E_2 \leq t_0$.

Observamos que, pelo fato de trabalharmos em $\mathbb{F}_2$, a soma do par de duas entradas erradas consecutivas x_i e x_{i+1} ou é igual a 0, se $x_i = x_{i+1}$, ou igual a 1, se $x_i \neq x_{i+1}$. No entanto, isto também acontece, caso as entradas x_i e x_{i+1} estejam ambas corretas. Assim, o número de erros em y_S é E_1 e o número de erros em $(y_L, y_R^{(1)})$ é $E_1 + 2E_2$, pois os erros de apenas uma entrada estão em y_L ou $y_R^{(1)}$ enquanto que os erros de duas entradas estão

nos dois vetores, ou seja, são contados duas vezes.

O seguinte resultado também será útil na validação do algoritmo de decodificação de erros de pares de símbolos.

Lema 3.3.2. *Se* $c \in \mathcal{C}$, $y = \pi(c) + e$ *e* $\omega_H(e) \leq t_0$, *então* $\mathcal{D}_{\mathcal{C}}(y_S) = c'$ *ou* $\mathcal{D}_{\mathcal{C}_2}((y_L, y_R^{(1)})) = c$.

Demonstração. Se $E_1 \leq t$, então o número de erros em y_S é no máximo t, que é a capacidade de correção do decodificador $\mathcal{D}_{\mathcal{C}}$, logo $\mathcal{D}_{\mathcal{C}}$ decodifica y_S corretamente. Caso contrário, se $E_1 \geq t+1$, como $E_1 + E_2 \leq t_0$, temos $E_2 \leq t_0 - (t+1)$, então o número de erros em $(y_L, y_R^{(1)})$ satisfaz

$$\begin{aligned} E_1 + 2E_2 &= (E_1 + E_2) + E_2 \leq t_0 + t_0 - (t+1) = 2\left\lceil \tfrac{3t+1}{2} \right\rceil - (t+1) \\ &\leq 2\left(\tfrac{3t+1}{2}\right) - (t-1) = 2t. \end{aligned}$$

Isso implica que o decodificador $\mathcal{D}_{\mathcal{C}_2}((y_L, y_R^{(1)}))$ é bem sucedido, isto é, $\mathcal{D}_{\mathcal{C}_2}((y_L, y_R^{(1)})) = c$. □

Pelo Lema 3.3.2, pelo menos um dos decodificadores $\mathcal{D}_{\mathcal{C}}$ e $\mathcal{D}_{\mathcal{C}_2}$ é bem sucedido. Entretanto, não é óbvio qual deles o é e a principal tarefa do algoritmo "subjascente", a aplicação decodificadora $\mathcal{D}_\pi$, que agora descrevemos, é identificar qual dos decodificadores é bem sucedido.

Seja y um vetor recebido, a saída do decodificador $\mathcal{D}_\pi(y) = \hat{c}$ é calculada do seguinte modo:

Algoritmo de Decodificação 3 ($\mathcal{D}_\pi$) :

Passo 1: $c_1 = \mathcal{D}_\mathcal{C}(y_S)$, $e_1 = d_H(c_1, y_S)$.

Passo 2: $c_2 = \mathcal{D}_{\mathcal{C}_2}((y_L, y_R^{(1)}))$, $e_2 = d_H((c_2, c_2), (y_L, y_R^{(1)}))$.

Passo 3: Se $c_2 = F$ então $\hat{c} = c_1^*$.

Passo 4: Se $c_1 = F$ ou $\omega_H(c_1)$ é ímpar, então $\hat{c} = c_2$.

Passo 5: Se $e_1 \leq \lfloor \frac{t+2}{2} \rfloor$, então $\hat{c} = c_1^*$.

Passo 6: Se $e_1 > \lfloor \frac{t+2}{2} \rfloor$, seja $e_1 = \lfloor \frac{t+2}{2} \rfloor + a$, $(1 \leq a \leq \lceil \frac{t}{2} \rceil - 1)$

a) Se $e_2 \leq t_0 + a$, então $\hat{c} = c_2$.

b) Caso contrário, $\hat{c} = c_1^*$.

A corretabilidade do decodificador é provada no próximo Teorema.

Teorema 3.3.3. *As saídas do decodificador satisfazem* $\mathcal{D}_\pi(y) = \hat{c} = c$.

Demonstração. De acordo com o Lema 3.3.2, pelo menos um dos dois decodificadores nos Passos 1 e 2 é bem sucedido. Passos 3-6 ajudam a determinar qual dos dois decodificadores funciona.

Passo 3: Pelo Lema 3.3.2, se $c_2 = F$ então $\hat{c} = c_1^*$. Se passarmos para os próximos passos significa que c_2 é uma palavra do código.

Passo 4: Como y_S é uma versão com ruído da palavra c', a operação de decodificação no Passo 1 tenta decodificar c', que tem peso par como visto na Observação 2.4.2. Se $c_1 = F$ ou o peso de Hamming de c_1 for ímpar, então esta operação de decodificação falhará, implicando que a operação de decodificação na Passo 2 foi bem sucedido. Se chegarmos aos Passos 5 e 6, então $\omega_H(c_1)$ deve ser par.

Passo 5: Se $e_1 \leq \lfloor \frac{t+2}{2} \rfloor$ e a palavra c_1 obtida no Passo 1 tem $d_H(c_1, c) > 1$, então a palavra y_S será decodificada incorretamente para uma palavra do código de peso par. Afirmamos que, como os pesos de c' e c_1 são pares, então $d_H(c', c_1)$ é par. De fato, sejam

$$\begin{aligned} A &= \{i;\ (c_1)_i = c'_i = 1\}, \\ B &= \{i;\ (c_1)_i = 1 \ e\ c'_i = 0\}, \\ C &= \{i;\ (c_1)_i = 0 \ e\ c'_i = 1\}. \end{aligned}$$

Note que $\mid A \mid + \mid B \mid = \omega_H(c_1)$ e $\mid A \mid + \mid C \mid = \omega_H(c')$, ou seja, $\mid A \mid + \mid B \mid$ e $\mid A \mid + \mid C \mid$ são números inteiros pares. Além disso, $\mid B \mid + \mid C \mid = d_H(c', c_1)$. Suponha que $d_H(c', c_1)$ seja ímpar. Então $\mid B \mid$ é ímpar ou $\mid C \mid$ é ímpar. Suponha, sem perda de generalidade, que $\mid B \mid$ seja ímpar. Como $\mid A \mid + \mid B \mid$ é par, então $\mid A \mid$ é ímpar. Daí, como $\mid A \mid + \mid C \mid$ é par, então $\mid C \mid$ é ímpar. Logo, $d_H(c', c_1) = \mid B \mid + \mid C \mid$ é par (pois a soma de números ímpares é par), contradizendo a hipótese.

Se a palavra enviada foi c e, lembrando que a distância mínima do código $\mathcal{C}$ é $2t+1$, então, pela desigualdade triangular, o número de erros reais em y_S, isto é, $d_H(c', y_S)$, satisfaz

$$d_H(c', y_S) + d_H(c_1, y_S) \geq d_H(c', c_1) \geq 2t + 2.$$

Como $E_1 = d_H(c', y_S)$ e $e_1 = d_H(c_1, y_S)$, temos

$$\begin{aligned} E_1 &\geq 2t + 2 - e_1 \geq 2t + 2 - \left\lfloor \frac{t+2}{2} \right\rfloor \geq 2t + 2 - \left(\frac{t+2}{2} \right) \\ &= \frac{3t}{2} + 1 = \frac{3t+1}{2} + \frac{1}{2} \geq \left\lfloor \frac{3t+1}{2} \right\rfloor + \frac{1}{2} = t_0 + \frac{1}{2} > t_0 \end{aligned}$$

contradizendo o fato do número de erros em y_S ser no máximo t_0. Assim, a condição em e_1 implica que a operação de decodificação $c_1 = \mathcal{D}_{\mathcal{C}}(y_S) = c'$ é bem sucedida e, pelo Lema 3.3.1, podemos concluir

$$\hat{c} = c_1^* = c.$$

Passo 6: Resta verificar o caso em que $e_1 > \lfloor \frac{t+2}{2} \rfloor$. Como $e_1 \leq t$, podemos escrever $e_1 = \lfloor \frac{t+2}{2} \rfloor + a$, com $1 \leq a \leq \lceil \frac{t}{2} \rceil - 1$.

Assuma que o decodificador do Passo 2 falhe. Pelo Lema 3.3.2, a decodificação $c_1 = \mathcal{D}_{\mathcal{C}}(y_S)$ é bem sucedida, implicando

$$E_1 = e_1 = \left\lfloor \frac{t+2}{2} \right\rfloor + a.$$

Assim, como $E_1 + E_2 \leq t_0$, o valor de E_2 satisfaz

$$\begin{aligned} E_2 &\leq t_0 - E_1 = t_0 - \left\lfloor \frac{t+2}{2} \right\rfloor - a \leq \frac{3t+1}{2} - \left\lfloor \frac{t}{2} \right\rfloor - 1 - a \\ &= \frac{3t+1}{2} + \left\lceil \frac{-t}{2} \right\rceil - 1 - a < \frac{3t+1}{2} + \left(\frac{-t}{2} + 1 \right) - 1 - a \\ &= t + \frac{1}{2} - a. \end{aligned}$$

Como $E_2 \in \mathbb{Z}$, segue $E_2 \leq t - a$, já que $t - a \in \mathbb{Z}$.

O número total de erros em $(y_L, y_R^{(1)})$ é

$$E_1 + 2E_2 = (E_1 + E_2) + E_2 \leq t_0 + t - a = \left\lfloor \frac{3t+1}{2} \right\rfloor + t - a = \left\lfloor \frac{5t+1}{2} \right\rfloor - a. \quad (3.11)$$

Como o decodificador $\mathcal{D}_{\mathcal{C}_2}((y_L, y_R^{(1)}))$ falha e a distância mínima de $\mathcal{C}_2$ é $2d_H(\mathcal{C}) = 4t+2$, se c é a palavra enviada, pela desigualdade triangular, temos

$$d_H((c_2, c_2), (y_L, y_R^{(1)})) + d_H((c, c), (y_L, y_R^{(1)})) \geq d_H((c, c), (c_2, c_2)) \geq 4t + 2. \quad (3.12)$$

Daí, como $e_2 = d_H((c_2, c_2), (y_L, y_R^{(1)}))$ e $E_1 + 2E_2 = d_H((c, c), (y_L, y_R^{(1)}))$ segue que o peso e_2 do vetor erro no Passo 2 deve satisfazer, por (3.11) e (3.12),

$$\begin{aligned} e_2 &\geq 4t + 2 - (E_1 + 2E_2) \geq 4t + 2 - \left(\left\lfloor \frac{5t+1}{2} \right\rfloor - a \right) \\ &\geq 4t + 2 - \left(\frac{5t+1}{2} \right) + a = \frac{3t+1}{2} + a + 1 \\ &\geq \lfloor \frac{3t+1}{2} \rfloor + a + 1 = t_0 + a + 1. \end{aligned}$$

Logo, se $e_2 < t_0 + a + 1$, isto é, $e_2 \leq t_0 + a$, então o decodificador do Passo 2 deve funcionar, daí $\hat{c} = c_2$.

Agora, suponha que a decodificação na Etapa 1 falhe. Como no Passo 4, isso significa que o números de erros E_1 em y_S é no mínimo

$$\begin{aligned} E_1 &\geq 2t + 2 - e_1 = 2t + 2 - \left(\left\lfloor \frac{t+2}{2} \right\rfloor + a \right) \geq 2t + 2 - \left(\frac{t+2}{2} \right) - a \\ &= \frac{3t+2}{2} - a \geq \left\lfloor \frac{3t+1}{2} \right\rfloor + \frac{1}{2} - a \geq t_0 - a. \end{aligned}$$

Como $E_1 + E_2 \leq t_0$, então E_2 deve satisfazer $0 \leq E_2 \leq t_0 - (t_0 - a) = a$. Logo, a operação de decodificação $\mathcal{D}_{\mathcal{C}_2}((y_L, y_R^{(1)}))$ é bem sucedida e $E_1 + 2E_2$, o número total de erros em $(y_L, y_R^{(1)})$, deve satisfazer

$$E_1 + 2E_2 = (E_1 + E_2) + E_2 \leq t_0 + a.$$

Assim, se $e_2 = E_1 + 2E_2 > t_0 + a$, então o decodificador no Passo 1 deve ser bem sucedido. Isso completa a explicação das atribuições em a) e b) no Passo 5. □

Vamos aplicar o algoritmo de decodificação no seguinte exemplo.

Exemplo 3.3.4. *Seja $\mathcal{C}$ o código BCH cíclico, binário, de comprimento 15 e corretor de 3 erros, então $d_H(\mathcal{C}) = 7$, $d_P(\mathcal{C}) = 7 + \lceil \frac{7}{2} \rceil = 11$, $t = 3$ e $t_0 = 5$. Logo o código pode corrigir até 5 pares de erros. Assuma a palavra armazenada como a palavra com todas as coordenadas nulas $\mathbf{0}$.*

Seja y o vetor recebido

$$y = (00, 11, 10, 00, 00, 00, 11, 00, 10, 00, 00, 11, 00, 00, 00).$$

Assim,

$$\begin{aligned} y_L &= (0,1,1,0,0,0,1,0,1,0,0,1,0,0,0), \\ y_R &= (0,1,0,0,0,0,1,0,0,0,0,1,0,0,0), \\ y_S &= (0,0,1,0,0,0,0,0,1,0,0,0,0,0,0), \end{aligned}$$

e

$$y_R^{(1)} = (0,0,1,0,0,0,0,1,0,0,0,0,1,0,0).$$

No Passo 1 do algoritmo de decodificação, calculamos

$$c_1 = \mathcal{D}_{\mathcal{C}}(y_S),\ e_1 = d_H(c_1, y_S).$$

Como y_S sofre 2 erros e o decodificador $\mathcal{D}_{\mathcal{C}}$ pode decodificar no máximo três erros, temos $c_1 = \mathbf{0}$ e $e_1 = 2$. No Passo 2, calculamos

$$c_2 = \mathcal{D}_{\mathcal{C}_2}((y_L, y_R^{(1)})),\ e_2 = d_H((c_2, c_2), (y_L, y_R^{(1)})).$$

A palavra $(y_L, y_R^{(1)})$ sofreu oito erros e como o decodificador $\mathcal{C}_2$ tem distância mínima 14, o decodificador $\mathcal{D}_{\mathcal{C}_2}$ pode corrigir até seis erros. Logo, a saída c_2 é igual à F, indicando que não existe uma palavra de distância no máximo seis de $(y_L, y_R^{(1)})$, ou há alguma palavra c_2 tal que $e_2 = 6$. A condição no Passo 4 falha, mas a condição no Passo 5 é satisfeita, pois $e_1 = 2 = \lfloor \frac{3+2}{2} \rfloor = 2$. Portanto, concluímos que o decodificador no Passo 1 é bem sucedido e podemos decodificar a palavra com $\hat{c} = \mathbf{0}^* = \mathbf{0}$. Note que a operação $\mathbf{0}^*$ pode resultar em $\mathbf{0}$ ou $\mathbf{1}$, mas podemos eliminar a palavra $\mathbf{1}$ pois sua distância da palavra recebida é muito grande.

Como outro exemplo, considere o vetor recebido y dado por

$$y = (10, 00, 01, 00, 10, 00, 00, 00, 00, 10, 00, 00, 00, 10, 00),$$

com vetores associados

$$\begin{aligned} y_L &= (1,0,0,0,0,0,0,0,0,1,0,0,0,1,0), \\ y_R &= (0,0,1,0,0,0,0,0,0,0,0,0,0,0,0), \\ y_S &= (1,0,1,0,1,0,0,0,0,1,0,0,0,1,0), \\ y_R^{(1)} &= (0,0,0,1,0,0,0,0,0,0,0,0,0,0,0). \end{aligned}$$

A palavra y_S sofreu cinco erros, então no Passo 1 a palavra decodificada é c_1 ou é o símbolo falho F ou alguma palavra de peso sete ou oito com distância igual a dois ou três de y_S, respectivamente. Vamos assumir para esse exemplo que c_1 é uma palavra de peso oito e $e_1 = 3$.

A palavra de entrada $(y_L, y_R^{(1)})$ sofreu cinco erros. Assim, no Passo 2, o decodificador $\mathcal{D}_{\mathcal{C}_2}((y_L, y_R^{(1)}))$ será bem sucedido, logo $c_2 = 0$ e $e_2 = 5$. Agora, as condições dos Passos 4 e 5 não são satisfeitas, então o Passo 6 vai determinar qual decodificador vai funcionar. Primeiramente, vemos que $a = 1$ e então $e_2 = 5 < 5 + 1 = t_0 + a$. Assim, a condição no Passo 6 a) é satisfeita e concluímos que o segundo decodificador é bem sucedido, isto é, $\hat{c} = 0$.

Exemplo 3.3.5. *Seja C o código do Exemplo* (3.2.18)*, então o Decodificador 3 pode corrigir até $t_0 = \left\lceil \frac{3t_H+1}{2} \right\rceil = 2$. Os vetores de peso ≤ 1 com suas respectivas síndromes estão relacionados na tabela abaixo*

líder	*síndrome*
(0000000)	*(000)*
(1000000)	*(100)*
(0100000)	*(010)*
(0010000)	*(001)*
(0001000)	*(110)*
(0000100)	*(011)*
(0000010)	*(111)*
(0000001)	*(101)*

Temos ainda que o código de dupla repetição $\mathcal{C}_2 = (\mathcal{C}, \mathcal{C})$ *tem matriz teste de paridade*

$$H_2 = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 \end{pmatrix},$$

distância mínima 6 e capacidade de correção 2. Os vetores de peso ≤ 2 *e suas respectivas síndromes estão relacionadas na tabela abaixo*

líder	*síndrome*
(00000000000000)	(0000000000)
(10000000000000)	(1000000000)
(01000000000000)	(0100000000)
(00100000000000)	(0010000000)
(00010000000000)	(0001000000)
(00001000000000)	(0000100000)
(00000100000000)	(0000010000)
(00000010000000)	(0000001000)
(00000001000000)	(0000000100)
(00000000100000)	(0000000010)
(00000000010000)	(0000000001)
(00000000001000)	(1101000110)
(00000000000100)	(0110100011)
(00000000000010)	(1110010111)
(00000000000001)	(1010001101)
(11000000000000)	(1100000000)
(10100000000000)	(1010000000)
(10010000000000)	(1001000000)
(10001000000000)	(1000100000)
(10000100000000)	(1000010000)
(10000010000000)	(1000001000)
(10000001000000)	(1000000100)
(10000000100000)	(1000000010)
(10000000010000)	(1000000001)
(10000000001000)	(0101000110)
(10000000000100)	(1110100011)
(10000000000010)	(0110010111)
(10000000000001)	(0010001101)
(01100000000000)	(0110000000)
(01010000000000)	(0101000000)
(01001000000000)	(0100100000)
(01000100000000)	(0100010000)
(01000010000000)	(0100001000)
(01000001000000)	(0100000100)
(01000000100000)	(0100000010)

líder	*síndrome*
(01000000010000)	(0100000001)
(01000000001000)	(1001000110)
(01000000000100)	(0010100011)
(01000000000010)	(1010010111)
(01000000000001)	(1110001101)
(00110000000000)	(0011000000)
(00101000000000)	(0010100000)
(00100100000000)	(0010010000)
(00100010000000)	(0010001000)
(00100001000000)	(0010000100)
(00100000100000)	(0010000010)
(00100000010000)	(0010000001)
(00100000001000)	(1111000110)
(00100000000100)	(0100100011)
(00100000000010)	(1100010111)
(00100000000001)	(1000001101)
(00011000000000)	(0001100000)
(00010100000000)	(0001010000)
(00010010000000)	(0001001000)
(00010001000000)	(0001000100)
(00010000100000)	(0001000010)
(00010000010000)	(0001000001)
(00010000001000)	(1100000110)
(00010000000100)	(0111100011)
(00010000000010)	(1111010111)
(00010000000001)	(1011001101)
(00001100000000)	(0000110000)
(00001010000000)	(0000101000)
(00001001000000)	(0000100100)
(00001000100000)	(0000100010)
(00001000010000)	(0000100001)
(00001000001000)	(1101100110)
(00001000000100)	(0110000011)
(00001000000010)	(1110110111)
(00001000000001)	(1010101101)

líder	*síndrome*
(00000110000000)	(0000011000)
(00000101000000)	(0000010100)
(00000100100000)	(0000010010)
(00000100010000)	(0000010001)
(00000100001000)	(1101010110)
(00000100000100)	(0110110011)
(00000100000010)	(1110000111)
(00000100000001)	(1010011101)
(00000011000000)	(0000001100)
(00000010100000)	(0000001010)
(00000010010000)	(0000001001)
(00000010001000)	(1101001110)
(00000010000100)	(0110101011)
(00000010000010)	(1110011111)
(00000010000001)	(1010000101)
(00000001100000)	(0000000110)
(00000001010000)	(0000000101)
(00000001001000)	(1101000010)
(00000001000100)	(0110100111)
(00000001000010)	(1110010011)
(00000001000001)	(1010001001)
(00000000110000)	(0000000011)
(00000000101000)	(1101000100)
(00000000100100)	(0110100001)
(00000000100010)	(1110010101)
(00000000100001)	(1010001111)
(00000000011000)	(1101000111)
(00000000010100)	(0110100010)
(00000000010010)	(1110010110)
(00000000010001)	(1010001100)
(00000000001100)	(1011100101)
(00000000001010)	(0011010001)
(00000000001001)	(0111001011)
(00000000000110)	(1000110100)
(00000000000101)	(1100101110)
(00000000000011)	(0100011010)

Suponha que seja recebida a palavra

1. $y_1 = [(0,1),(1,0),(0,1),(1,0),(0,0),(0,0),(0,1)]$. *Então*

   $y_L = (0101000)$ $\qquad$ $y_S = y_L + y_R = (1111001)$

   $y_R = (1010001)$ $\qquad$ $y_R^{(1)} = (1101000)$

   Passo 1: $H \cdot y_S^T = (100)$, *logo* $e_1 = (1000000)$ *e, portanto,* $\mathcal{D}_\mathcal{C}(y_S) = c_1 = y_S - e_1 = (0111001)$ *e* $e_1 = d_H(c_1, y_S) = 1$.

   Passo 2: $H_2 \cdot (y_L, y_R^{(1)})^T = (1000000000)$, *logo* $e_2' = (10000000000000)$. *Assim,* $(c_2, c_2) = (11010001101000)$ *e* $e_2 = d_H((c_2, c_2), (y_L, y_R^{(1)})) = 1$.

   Passo 4: $\omega_H(c_1) = 4$ *é par, logo passamos para o próximo passo.*

   Passo 5: $e_1 = 1 \leq \left\lceil \frac{t_H+2}{2} \right\rceil = 1$, *então* $\hat{c} = c_1^*$.

Além disso, $\tilde{c} = (0010111)$ *e* $\tilde{c}+\mathbf{1} = (1101000)$. *Daí* $d_H(y_L, \tilde{c}) = 6$ *e* $d_H(y_L, \tilde{c}+\mathbf{1}) = 1$, *logo* $d_H(y_L, \tilde{c}) > d_H(y_L, \tilde{c}+\mathbf{1})$ *e, portanto,* $\hat{c} = \tilde{c}+\mathbf{1} = (1101000)$.

2. $y_2 = [(0,0),(1,1),(0,1),(1,1),(1,1),(0,1),(1,0)]$. *Então*

   $y_L = (0101101)$ $\quad$ $y_S = y_L + y_R = (0010011)$
   $y_R = (0111110)$ $\quad$ $y_R^{(1)} = (0011111)$

   Passo 1: $H \cdot y_S^T = (011)$, *logo* $e_1 = (0000100)$ *e, portanto,* $\mathcal{D}_{\mathcal{C}}(y_S) = c_1 = y_S - e_1 = (0010111)$ *e* $e_1 = d_H(c_1, y_S) = 1$.

   Passo 2: $H_2 \cdot (y_L, y_R^{(1)})^T = (0010010110)$ *não se encontra na tabela, logo* $c_2 = F$.

   Passo 3: Como $c_2 = F$, $\hat{c} = c_1^*$. *Assim,* $\tilde{c} = (0001101)$ *e* $\tilde{c}+\mathbf{1} = (1110010)$ *e daí* $d_H(y_L, \tilde{c}) = 1$ *e* $d_H(y_L, \tilde{c}+\mathbf{1}) = 6$. *Logo* $d_H(y_L, \tilde{c}) < d_H(y_L, \tilde{c}+\mathbf{1})$ *e, portanto,* $\hat{c} = \tilde{c} = (0001101)$.

3. $y_3 = [(0,0),(0,1),(0,1),(1,1),(1,0),(0,0),(1,0)]$, *então*

   $y_L = (0001101)$ $\quad$ $y_S = y_L + y_R = (0110101)$
   $y_R = (0111000)$ $\quad$ $y_R^{(1)} = (0011100)$

   Passo 1: $H \cdot y_S^T = (101)$, *logo* $e_1' = (0000001)$ *e, portanto,* $\mathcal{D}_{\mathcal{C}}(y_S) = c_1 = y_S - e_1 = (0110100)$ *e* $e_1 = d_H(c_1, y_S) = 1$.

   Passo 2: $H_2 \cdot (y_L, y_R^{(1)})^T = (1010001100)$, *logo* $e_2' = (00000000010001)$. *Assim,* $(c_2, c_2) = (00011010001101)$ *e* $c_2 = (0001101)$.

   Passo 4: Como $\omega_H(c_1) = 3$ *é ímpar, temos* $\hat{c} = c_2$.

Para completar a apresentação do decodificador, retornamos para a construção do decodificador (3.9) Esse decodificador recebe dois vetores, $y_1 = (y_{1,0}, \ldots, y_{1,n-1})$ e $y_2 = (y_{2,0}, \ldots, y_{2,n-1})$. Cada um deles é uma versão com erros de alguma palavra $c \in \mathcal{C}$ e o objetivo é corrigir o total de $2t$ erros nos dois vetores.

Utilizamos no Exemplo 3.2.18 o algoritmo de decodificação por síndrome, entretanto existem outras formas de decodificar tais palavras. Por exemplo, defina o vetor $\tilde{y} = (\tilde{y_0}, \ldots, \tilde{y}_{n-1})$ tal que, para todo $0 \leq i \leq n-1$, $\tilde{y_i} = y_{1,i}$ se $y_{1,i} = y_{2,i}$ e, caso contrário, $\tilde{y_i} =?$ para indicar um apagamento. Se o número de erros em $\tilde{y}$ é τ e o número de apagamentos é ρ, então $2\tau + \rho \leq 2t = d_H(\mathcal{C}) - 1$, que está dentro da capacidade de correção de erros e exclusões de $\mathcal{C}$. Fica apenas com o problema de definir um decodificador que corrija erros e exclusões para códigos cíclicos. Para isso, Yaacobi, Bruck e Siegel [28] referem à [21],[24]. Alternativamente, podemos tratar o código $\mathcal{C}_2$ como um código "concatenado", no qual o código interno é simplesmente um código de repetição de comprimento dois. Uma técnica geral para decodificar códigos concatenados é descrita na Seção 1.9

Um erro de pares de símbolos pode alterar o valor de uma única coordenada ou de ambas em um par de símbolos. No entanto, o conhecimento sobre o número máximo de erros de pares de símbolos de cada tipo pode ser conhecido antecipadamente. Por exemplo, um erro de pares de símbolos que corrompe apenas uma coordenada pode ser o resultado de uma coordenada que foi gravada erroneamente na mídia, enquanto um erro de duas coordenadas pode ser resultado de um ruído de leitura. Assim, consideramos códigos que distinguem entre esses dois tipos de erros. Especificamente, dizemos que um código é um *código corretor de* (t_1, t_2)*-erros de pares de símbolos* se puder corrigir até t_1 erros de pares de símbolos de apenas uma coordenada e até t_2 erros de pares de duas coordenadas.

Nossa discussão de código corretor de (t_1, t_2)-erros de pares de símbolos usa como ponto de partida um código linear cíclico $\mathcal{C}$ com distância mínima de Hamming $d_H(\mathcal{C})$ e um decodificador $\mathcal{D}_\mathcal{C}$. O próximo teorema fornece uma condição em $d_H(\mathcal{C})$ que implica que o código $\mathcal{C}$ é um código corretor de (t_1, t_2)-erros de pares de símbolos.

Teorema 3.3.6. *Um código cíclico linear* $\mathcal{C}$ *de comprimento* n *e distância mínima* $d_H(\mathcal{C})$ *é um código corretor de* (t_1, t_2)*-erros de pares de símbolos, se*

$$d_H(\mathcal{C}) \geq min\{t_1 + 2t_2 + 1, 2t_1 + 1\},$$

e $t_1 + t_2 < n/2$.

Demonstração. Examinaremos duas abordagens diferentes para corrigir tais erros.

1. A primeira abordagem usa o decodificador $\mathcal{D}_{\mathcal{C}_2}$ do código de dupla repetição de $\mathcal{C}$ no vetor $(y_L, y_R^{(1)})$. Esse decodificador vai precisar corrigir $t_1 + 2t_2$ erros e assim

   $$2d_H(\mathcal{C}) = d_H(\mathcal{C}_2) \geq 2(t_1 + 2t_2) + 1,$$

   ou

   $$d_H(\mathcal{C}) \geq t_1 + 2t_2 + 1,$$

   pois $d_H(\mathcal{C}) \in \mathbb{Z}$.

2. A segunda abordagem usa o decodificador $\mathcal{D}_\mathcal{C}$ aplicado ao vetor y_S. Esse decodificador deve corrigir t_1 erros e, assim, $d_H(\mathcal{C}) = 2t_1 + 1$. Pelo Lema 3.3.1, a condição $t_1 + t_2 \leq t_0 < n/2$ garante o recuperação total da palavra armazenada c baseada na decodificação de c'.

Concluímos que se a condição na afirmação do Teorema é satisfeita, podemos escolher qualquer uma destas abordagens como base para um decodificador bem sucedido. □

Do Teorema 3.3.6, concluímos que conhecer os valores t_1 e t_2 simplifica significantemente a operação de decodificação, desde que saibamos qual dos dois decodificadores usar para corrigir os pares de erros. No entanto, este conhecimento pode reduzir o limite inferior na distância mínima de Hamming necessária do código $\mathcal{C}$ e, portanto, a redundância de código necessária. Para ver isto, assuma que construímos um código corretor de (t_1, t_2)-erros de pares de símbolos usando um código linear cíclico $\mathcal{C}$ que corrige $t_1 + t_2$ pares de erros. Assim, de acordo com o Teorema 2.4.3 e a Proposição 2.1.24, sua distância mínima de Hamming $d_H(\mathcal{C})$ deve satisfazer

$$d_H(\mathcal{C}) + \lceil \frac{d_H(\mathcal{C})}{2} \rceil \geq 2(t_1 + t_2) + 1. \tag{3.13}$$

Por outro lado, de acordo com o Teorema 3.3.6, $d_H(\mathcal{C}) \geq \min\{t_1 + 2t_2 + 1, 2t_1 + 1\}$. Como $2(t_1 + t_2) + 1 \geq t_1 + 2t_2 + 1$ e $2(t_1 + t_2) + 1 \geq 2t_1 + 1$, para quaisquer dois inteiros não negativos t_1 e t_2, este limite inferior em $d_H(\mathcal{C})$ não é maior que o limite inferior em $d_H(\mathcal{C})$ implicado por (3.13).

Para concluir a discussão dessa seção, notemos que o decodificador $\mathcal{D}_\pi$ apresentado usa dois decodificadores: o primeiro é $\mathcal{D}_\mathcal{C}$ para o código $\mathcal{C}$ e o segundo é $\mathcal{D}_{\mathcal{C}_2}$ para o código $\mathcal{C}_2$. Como não há nenhum requisito específico para esses decodificadores, além da capacidade de correção de erros e eliminação, podemos usar qualquer um dos decodificadores existentes para códigos cíclicos. Portanto, a complexidade do decodificador $\mathcal{D}_\pi$ será da mesma ordem que os melhores decodificadores para códigos cíclicos. Yaakobi, Bruck e Siegel afirmam em [28] que isso significa uma melhora significativa no decodificador do tipo síndrome com relação ao complexidade do decodificador. Porém, o decodificador $\mathcal{D}_\pi$ pode corrigir até $t_0 = \lfloor (3t + 1)/2 \rfloor$ erros de pares, para um código cíclico linear com distância mínima de Hamming $d_H = 2t + 1$, que é menor que a capacidade de correção do decodificador do tipo síndrome para códigos lineares, como afirmado por Hirotomo, Takita e Morii [14]. De fato, $\mathcal{C}$ corrige até t_P erros de pares se, e somente se, $d_P(\mathcal{C}) \geq 2t_P + 1$, assim

$$t_P \leq \frac{d_P(\mathcal{C}) - 1}{2} \leq \frac{2d_H - 1}{2} = \frac{4t + 1}{2} \Rightarrow t_P \leq \left\lfloor \frac{4t + 1}{2} \right\rfloor = 2t + \left\lfloor \frac{1}{2} \right\rfloor = 2t.$$

Logo o algoritmo de decodificação 2 pode corrigir até $2t$ erros de pares. Além disso, para $t = 0$ ou 1, $t_0 = 2t$ e, para $t \geq 2$, $t_0 < 2t$, isto é, para $t \geq 2$ a capacidade de correção do decodificador $\mathcal{D}_\pi$ é estritamente menor que o decodificador do tipo síndrome.

Capítulo 4

Códigos para canais de leitura de b-símbolos

Sejam $\mathcal{C}$ um (n, d)-código sobre o alfabeto $\mathbb{F}_q$ e $b \in \mathbb{N}$, com $3 \leq b < n$. Neste capítulo, estudamos os canais de leitura de b-símbolos, nos quais b símbolos são detectados em cada operação de leitura como apresentado por Yaakobi, Bruck e Siegel em [28] e [29]. Primeiramente, definimos formalmente o modelo de canal de leitura de b-símbolos. Provamos alguma de suas propriedades básicas, de forma análoga à feita no modelo de canal de leitura de pares de símbolos do Capítulo 2. Em seguida, nos voltamos para as construções de códigos para o canal de leitura de b-símbolos. Provamos a cota de Singleton e definimos os códigos de b-símbolos MDS (*Maximum Distance Separable*), como feito por Ding, Zhang e Ge [7]. Generalizando os resultados do Capítulo 2, analisamos as propriedades de distância de b-símbolos dos códigos obtidos pela intercalação dos componentes de b códigos e após descrevemos um algoritmo de decodificação que decodifica até a capacidade de correção de um código. Finalmente, estudamos as propriedades de duas famílias específicas de códigos de b-símbolos, os códigos correspondentes a todo espaço $\mathcal{A}^n$, isto é, $\mathcal{C} = \mathcal{A}^n$ e os códigos de Hamming cíclicos lineares de comprimento $n = 2^m - 1,\ m \geq 3$.

4.1 Propriedades Básicas

Definição 4.1.1. *Para $b \geq 3$, o* **vetor de b-símbolos** *correspondente ao vetor $x = (x_0, x_1, \ldots, x_{n-1}) \in \mathcal{A}^n$ é definido como*

$$\pi_b(x) = [(x_0, \ldots, x_{b-1}), \ldots, (x_{n-1}, x_0, \ldots, x_{b-2})] \in \left(\mathcal{A}^b\right)^n.$$

Nos referimos a cada coordenada de $\pi_b(x)$ como *b-símbolos*. Nos referimos à i-ésima coordenada do vetor $\pi_b(x)$ com $\pi_b(x)_i = (x_i, x_{i+1}, \ldots, x_{i+b-1})$. Por conveniência de notação, se $\pi_b(x)_i$ não é a b-upla nula, escrevemos $\pi_b(x)_i \neq \mathbf{0}$.

Definição 4.1.2. *O* **código de leitura de b-símbolos** *de um código $\mathcal{C}$ é*

$$\pi_b(\mathcal{C}) = \{\pi_b(c); c \in \mathcal{C}\}.$$

Definição 4.1.3. *A* **distância de b-símbolos** *entre x e y, denotada por $d_b(x, y)$, é o número inteiro*

$$d_b(x, y) = d_H(\pi_b(x), \pi_b(y)) = \mid \{i;\ \pi_b(x)_i \neq \pi_b(y)_i\} \mid .$$

Por simplicidade, nos referimos a esta distância como **b-distância**.

Proposição 4.1.4. *Dados $u, v, w \in \mathcal{A}^n$, valem as seguintes propriedades:*

i) Positividade: $d_b(x, y) \geq 0$; com $d_b(x, y) = 0 \ \Leftrightarrow x = y$.

ii) Simetria: $d_b(x, y) = d_b(y, x)$.

iii) Desigualdade Triangular: $d_b(x, y) \leq d_b(x, w) + d_b(w, y)$.

Demonstração. Os dois primeiros itens são trivias. Para provar a desigualdade triangular observe que se $\pi_b(x)_i \neq \pi_b(y)_i$, para algum $i = 0, ..., n-1$, então soma-se um à $d_b(x, y)$ e pelo menos um à $d_b(x, w) + d_b(w, y)$, pois, caso contrário, se somassemos zero à $d_b(x, w) + d_b(w, y)$, teríamos $\pi_b(x)_i = \pi_b(w)_i$ e $\pi_b(w)_i = \pi_b(y)_i$, o que implicaria $\pi_b(x)_i = \pi_b(y)_i$, contradizendo a hipótese. Logo, $d_b(x, y) \leq d_b(x, w) + d_b(w, y)$. □

Pela Proposição 4.1.4, tem-se que a b-distância é uma métrica.

Definição 4.1.5. *A* **b-distância mínima** *de um código $\mathcal{C}$, denotada por $d_b(\mathcal{C})$, é definida como*

$$d_b = d_b(\mathcal{C}) = min\{d_b(x, y) \mid \ x, y \in \mathcal{C} e\ x \neq y\}.$$

Em geral, um código $\mathcal{C}$ sobre $\mathcal{A}^n$ de comprimento n, tamanho M e b-distância mínima d_b é chamado de $(n, M, d_b)_q$-código de b símbolos, com $q = \mid \mathcal{A}^n \mid$.

Definição 4.1.6. *Dado $x \in \mathcal{A}^n$, definimos o* **b-peso** *de um vetor x como*

$$\omega_b(x) = \omega_H(\pi_b(x)) = \mid \{i;\ \pi_b(x)_i \neq \mathbf{0}\} \mid .$$

Em outras palavras, $\omega_b(x) = d_b(x, \mathbf{0})$, para todo $x \in \mathcal{A}^n$.

Definição 4.1.7. *O* b**-peso** *de um código linear* $\mathcal{C}$ *é o inteiro*

$$\omega_b(\mathcal{C}) = min\{\omega_b(x) \mid \ x \in \mathcal{C} \setminus \{\mathbf{0}\}\}.$$

Proposição 4.1.8. *Seja* $\mathcal{C} \subset \mathcal{A}^n$ *um código linear, temos*

(i) $d_b(x, y) = \omega_b(x - y)$, *para todos* $x, y \in \mathcal{A}^n$.

(ii) $d_b(\mathcal{C}) = \omega_b(\mathcal{C})$.

Demonstração. Primeiramente, temos

$$\begin{aligned} \pi_b(x)_i + \pi_b(y)_i &= (x_i, x_{i+1}, \ldots, x_{i+b-1}) + (y_i, y_{i+1}, \ldots, y_{i+b-1}) \\ &= (x_i + y_i, x_{i+1} + y_{i+1}, \ldots, x_{i+b-1} + y_{i+b-1}) = \pi_b(x + y)_i. \end{aligned}$$

O item *(i)* segue diretamente das definições de b-distância e de b-peso de vetores

$$\begin{aligned} d_b(x, y) &= d_H(\pi_b(x), \pi_b(y)) = \{i \mid \ \pi_b(x)_i \neq \pi_b(y_i)\} \\ &= \{i \mid \ \pi_b(x)_i - \pi_b(y)_i \neq \mathbf{0}\} \\ &= \{i \mid \ \pi_b(x - y)_i \neq \mathbf{0}\} \\ &= \omega_H(\pi_b(x - y)) = \omega_b(x - y). \end{aligned}$$

O item *(ii)* segue do fato que, para todos os vetores $x, y \in \mathcal{C}$ com $x \neq y$, tem-se $z = x - y \in \mathcal{C} \setminus \{\mathbf{0}\}$ pois $\mathcal{C}$ é um código linear. Assim,

$$\begin{aligned} d_b(\mathcal{C}) &= min\{d_b(x, y) \mid x, y \in \mathcal{C} \ e \ x \neq y\} = min\{\omega_b(x - y) \mid x, y \in \mathcal{C} \ e \ x \neq y\} \\ &= min\{\omega_b(z) \mid z \in \mathcal{C} \setminus \{\mathbf{0}\}\} = \omega_b(\mathcal{C}). \end{aligned}$$

□

Observação: Se $\mathcal{A} = \{0, 1\}$, então $d_b(x, y) = \omega_b(x + y)$, pois $-y = y$, para todo $y \in \mathcal{A}$.

A seguinte proposição é uma generalização natural da Proposição 2.1.10.

Proposição 4.1.9. *Seja* $x = (x_0, x_1, \ldots, x_{n-1}) \in \mathcal{A}^n$ *tal que* $0 < \omega_H(x) \leq n - (b - 1)$. *Então*

$$\omega_H(x) + b - 1 \leq \omega_b(x) \leq b \cdot \omega_H(x).$$

Demonstração. Seja $S = \{i;\ x_i \neq 0\}$, então $\mid S \mid = \omega_H(x)$. Como todo símbolo x_i aparece em b coordenadas de $\pi_b(x)$, o número de b-símbolos não nulos em $\pi_b(x)$ é no máximo $\mid S \mid \cdot b = \omega_H(x) \cdot b$, implicando no limite superior $\omega_b(x) \leq b \cdot \omega_H(x)$.

Para provar o limite inferior, consideramos dois casos. Primeiro, assuma que x não possua um conjunto com b ou mais zeros consecutivos. Então, para cada índice i, temos $\pi_b(x)_i \neq 0$ e, assim, $\omega_b(x) = n$, o que implica

$$\omega_H(x) + b - 1 \leq n - (b-1) + (b-1) = n = \omega_b(x).$$

Agora, considere o caso em que x contenha uma sequência de no mínimo b zeros consecutivos. Como, por hipótese $\omega_H(x) > 0$, x possui pelo menos uma coordenada não nula, daí podemos encontrar um índice k tal que $x_k = x_{k+1} = \ldots = x_{k+b-1} = 0$, e $x_{k+b} \neq 0$. Assim, para todo $k+1 \leq i \leq k+b-1$, temos $\pi_b(x)_i \neq \mathbf{0}$. Além disso, para todo i tal que $x_i \neq 0$, temos também $\pi_b(x)_i \neq \mathbf{0}$. Logo

$$\omega_b(x) \geq \omega_H(x) + b - 1.$$

□

O resultado a seguir é uma generalização do Corolário 2.1.11.

Corolário 4.1.10. *Se $\mathcal{C}$ é um código linear tal que $0 < d_H(\mathcal{C}) \leq n - (b-1)$, então*

$$d_H(\mathcal{C}) + b - 1 \leq d_b(\mathcal{C}) \leq b \cdot d_H(\mathcal{C}).$$

Demonstração. Existem $x, y \in \mathcal{C}$ com $x \neq y$ tais que $d_H(x,y) = d_H(\mathcal{C})$. Pela Proposição 4.1.8, $\omega_H(x-y) = d_H(x,y) = d_H(\mathcal{C})$. Como $\mathcal{C}$ é linear, $z = x - y \in \mathcal{C}$, logo $\omega_H(z) = d_H(\mathcal{C})$ o que implica $0 < \omega_H(z) \leq n - (b-1)$. Assim, pela Proposição 4.1.9, $\omega_H(z) + b - 1 \leq \omega_b(z) \leq b \cdot \omega_H(z)$ e, portanto,

$$d_H(\mathcal{C}) + b - 1 \leq d_b(\mathcal{C}) \leq b \cdot d_H(\mathcal{C}).$$

□

Proposição 4.1.11. *Para todo vetor não nulo $x = (x_0, x_1, \ldots, x_{n-1})$ em $\mathcal{A}^n$, com $0 < \omega_b(x) < n$, temos $\omega_{b+1}(x) \geq \omega_b(x) + 1$.*

Demonstração. É claro que $\omega_{b+1}(x) \geq \omega_b(x)$, pois se $\pi_b(x)_i = (x_i, x_{i+1}, \ldots, x_{i+b-1}) \neq \mathbf{0}$ então $\pi_{b+1}(x)_i = (x_i, x_{i+1}, \ldots, x_{i+b}) \neq \mathbf{0}$, com $0 \leq i \leq n-1$. Além disso, como $0 < \omega_b(x) < n$, existe $0 \leq j \leq n-1$ tal que $\pi_b(x)_j = (x_j, x_{j+1}, \ldots, x_{j+b-1}) = \mathbf{0}$ e $x_{j+b} \neq 0$. Assim, $\pi_{b+1}(x)_j = (x_j, x_{j+1}, \ldots, x_{j+b}) \neq \mathbf{0}$ e, portanto, $\omega_{b+1}(x) \geq \omega_b(x) + 1$. □

Exemplo 4.1.12.

$$\begin{array}{lll} \omega(1110000) = 3, & \omega_3(1110000) = 5, & \omega_4(1110000) = 6 \\ \omega(1100001) = 3, & \omega_3(1100001) = 5, & \omega_4(1100001) = 6 \\ \omega(1101000) = 3, & \omega_3(1101000) = 6, & \omega_4(1101000) = 7 \\ \omega(1010100) = 3, & \omega_3(1010100) = 7, & \omega_4(1010100) = 7 \end{array}$$

Observe que, quanto mais próximas as coordenadas não nulas de um vetor estão uma das outras, menor é seu b-peso. Assim, para um peso de Hamming fixo, o menor b-peso acontece quando todas as coordenadas não nulas estão em posições consecutivas ciclicamente.

De forma análoga à Teoria Clássica de Códigos, o próximo resultado limita o número de elementos de um código $\mathcal{C}$ com comprimento e b-distância mínima fixa.

Teorema 4.1.13 (Cota de Singleton)**.** *Sejam $q \geq 2$ e $b \leq d_b \leq n$. Se $\mathcal{C}$ é um $(n, M, d_b)_q$-código de b-símbolos, então $M \leq q^{n-d_b+b}$.*

Demonstração. Suponha que $\mathcal{C}$ é um (n, M, d_b)-código de b-símbolos, com $q \geq 2$ e $b \leq d_b \leq n$. Exclua as últimas $d_b - b$ coordenadas de todas as palavras de $\mathcal{C}$. Note que quaisquer $d_b - b$ coordenadas consecutivas contribuem em no máximo $d_b - 1$ à b-distância. Assim, os vetores resultantes de comprimento $n - d_b + b$ continuam distintos, pois a menor b-distância entre dois vetores de $\mathcal{C}$ é d_b. Como o número máximo de vetores de comprimento $n - d_b + b$ sobre $\mathbb{F}_q$ é q^{n-d_b+b}, temos $M \leq q^{n-d_b+b}$. □

Definição 4.1.14. *Um $(n, M, d_b)_q$-código linear de b-símbolos $\mathcal{C}$, com $M = q^{n-d_b+b}$ é chamado de* **código de b-símbolos MDS** *(Maximum Distance Separable).*

Teorema 4.1.15. *Um $(n, d_b)_q$-código de b símbolos MDS, com $d_b < n$, é também um $(n, d_b + 1)_q$-código de $b + 1$ símbolos MDS.*

Demonstração. Para todo $c \in \mathcal{C}$, $\omega_b(c) \geq d_b$, pela Proposição 4.1.11, $\omega_{b+1}(c) \geq d_b + 1$. Assim, $d_{b+1} \geq d_b + 1$. Daí $\mid \mathcal{C} \mid = q^{n-d_b+b} \geq q^{n-d_{b+1}+b+1}$. Logo, pelo Teorema 4.1.13, $\mathcal{C}$ é um código de b-símbolos MDS. Além disso, como $q^{n-d_b+b} = q^{n-d_{b+1}+b+1}$, temos $d_{b+1} = d_b + 1$. □

Do Teorema 4.1.15 é possível encontrar novas famílias de códigos de b-símbolos MDS a partir de cada família de códigos MDS apresentados nos artigos [5], [7], [8] e [16].

No artigo [7], Ding, Zhang e Ge fazem construções de códigos MDS de b-símbolos a partir da Geometria Projetiva e códigos constacíclicos.

A partir de agora utilizaremos apenas o alfabeto $\mathcal{A} = \{\mathbf{0}, \mathbf{1}\}$.

Nosso próximo objetivo é generalizar adequadamente o Lema 2.4.1, dando uma caracterização útil de $\omega_b(x)$, para $b \geq 3$ arbitrário. Para fazer isso, introduzimos, para todo $x \in \mathcal{A}^n$, um vetor auxiliar $\hat{x} \in \mathcal{A}^n$, obtido do seguinte modo: em toda sequência de $b-2$, ou menos, zeros consecutivos em x cada entrada será trocada por 1. Formalmente, definimos $\hat{x}$ como segue. Se $(x_i, x_{i+1}, \ldots, x_{i+k}, x_{i+k+1}) = (1, 0, \ldots, 0, 1)$, para algum $0 \leq i \leq n-1$ e $k \leq b-2$, então $\hat{x}_j = 1 - x_j = 1$, para $i+1 \leq j \leq i+k$. Para todos os outros valores de j, $\hat{x}_j = x_j$.

Exemplo 4.1.16. *Sejam $b = 4$ e $x = (0, 1, 1, 0, 0, 0, 1, 0)$. Então*

$$\hat{x} = (1, 1, 1, 0, 0, 0, 1, 1).$$

Note que a sequência de zeros consecutivos começa na posição 3 e tem comprimento $3 > b-2 = 2$, assim esses zeros não mudam em $\hat{x}$, enquanto que a sequência de zeros ciclicamente consecutivos, começando na posição 7, tem comprimento $2 = b-2$ e, portanto, esses são os zeros que mudam para 1.

Agora, nós afirmamos e provamos a generalização do Lema 2.4.1, para $b \geq 3$.

Lema 4.1.17. *Para qualquer $x \in \mathcal{A}^n$ e inteiro $b \geq 3$,*

$$\omega_b(x) = \omega_H(\hat{x}) + (b-1) \cdot \frac{\omega_H(\hat{x}')}{2}.$$

Demonstração. Vamos mostrar primeiro que $\omega_b(x) = \omega_b(\hat{x})$. As únicas posições j para as quais $\pi_b(x)_j$ e $\pi_b(\hat{x})_j$ diferem são as que $\pi_b(x)_j = (x_j, \ldots, x_{j+b-1})$ contém uma sequência de comprimento $k+2$ da forma

$$(x_i, x_{i+1}, \ldots, x_{i+k}, x_{i+k+1}) = (1, 0, \ldots, 0, 1),$$

com $k \leq b-2$. As entradas nulas $x_{i+1}, \ldots, x_{i+k}$ aparecem no j-ésimo símbolo de $\pi_b(x)$ para $i-b+2 \leq j \leq i+k$. Entretanto, como $x_i = x_{i+k+1} = 1$, nessa faixa de valores de j, vemos que ambos $\pi_b(x)_j \neq 0$ e $\pi_b(\hat{x})_j \neq 0$. Para todas as outras posições j, as coordenadas correspondentes de $\pi_b(x)_j$ e $\pi_b(\hat{x})_j$ são as mesmas e assim $\pi_b(x)_j \neq \mathbf{0}$ se, e somente se, $\pi_b(\hat{x})_j \neq \mathbf{0}$.

Agora, vamos determinar o valor de $\omega_b(\hat{x})$, fazendo uso do fato que qualquer sequência de zeros consecutivos em $\hat{x}$ tem comprimento no mínimo $b-1$, pois nas sequências menores

os zeros de x se tornam um em $\hat{x}$. Seja

$$\begin{aligned} S_0 &= \{i;\ \pi_b(\hat{x})_i \neq 0,\ \hat{x}_i = 1\}, \\ S_1 &= \{i;\ \pi_b(\hat{x})_i \neq 0,\ \hat{x}_i = 0,\ \hat{x}_{i+1} = 1\}, \\ &\vdots \\ S_{b-2} &= \{i;\ \pi_b(\hat{x})_i \neq 0,\ \hat{x}_i = \ldots = \hat{x}_{i+b-3} = 0,\ \hat{x}_{i+b-2} = 1\}, \\ S_{b-1} &= \{i;\ \pi_b(\hat{x})_i \neq 0,\ \hat{x}_i = \ldots = \hat{x}_{i+b-2} = 0,\ \hat{x}_{i+b-1} = 1\}, \end{aligned}$$

Claramente, $\omega_H(\hat{x}) =\mid S_0 \mid$ e, como $S_j \cap S_l = \emptyset$, para todo $0 \leq j < l \leq b-1$, também temos

$$\omega_b(\hat{x}) =\mid \bigcup_{i=0}^{b-1} S_i \mid = \sum_{i=0}^{b-1} \mid S_i \mid .$$

Agora, vamos mostrar que, para todo $2 \leq l \leq b-1$, $\mid S_1 \mid = \mid S_l \mid$. Se $i \in S_1$ então $(\hat{x}_i, \hat{x}_{i+1}) = (0,1)$. Como em $\hat{x}$ não existe sequência com menos que $b-1$ zeros consecutivos, temos

$$(\hat{x}_{i-(b-2)}, \ldots, \hat{x}_i, \hat{x}_{i+1}) = (0, \ldots, 0, 1).$$

Note que $\pi_b(x)_{i-(l-1)} \neq \mathbf{0}$, pois como $2 \leq l \leq b-1$, temos $i-(b-2) \leq i-(l-1) \leq i-1$. Assim $\hat{x}_{i-(l-1)} = \ldots = \hat{x}_i = 0$ e $\hat{x}_{i+1} = 1$, daí $i-(l-1) \in S_l$. Logo, $\mid S_l \mid \geq \mid S_1 \mid$. Para a inequação oposta, note se $i \in S_l$, $l \geq 1$, então

$$(\hat{x}_i, \hat{x}_{i+1}, \ldots, \hat{x}_{i+l-1}, \hat{x}_{i+l}) = (0, \ldots, 0, 1).$$

Assim, $(\hat{x}_{i+l-1}, \hat{x}_{i+l}) = (0,1)$, então $i+l-1 \in S_1$, implicando que $\mid S_1 \mid \geq \mid S_l \mid$. Logo, $\mid S_1 \mid = \mid S_l \mid$, para todo $2 \leq l \leq b-1$. Como em (2.2), o vetor $\hat{x}'$ é definido por

$$\hat{x}' = (\hat{x}_0 + \hat{x}_1, \ldots, \hat{x}_{n-1} + \hat{x}_0),$$

e, assim, como na prova do Lema 2.4.1, $\mid S_1 \mid = \frac{\omega_H(\hat{x}')}{2}$. Logo

$$\omega_b(x) = \omega_b(\hat{x}) = \sum_{i=0}^{b-1} \mid S_i \mid = \omega_H(\hat{x}) + (b-1) \cdot \frac{\omega_H(\hat{x}')}{2}.$$

□

Exemplo 4.1.18. *Sejam $b = 4$ e $x = (1,0,0,0,1,0,0,1)$. Então $\omega_H(x) = 3$,*

$$\pi_4(x) = (1000, 0001, 0010, 0100, 1001, 0011, 0110, 1100),$$

e, assim, $\omega_4(x) = 8$. *Note que* $\omega_H(x) + b - 1 = 6 \leq \omega_4(x) = 8 \leq 4 \cdot \omega_H(x) = 12$, *ou seja, a Proposição 4.1.9 é satisfeita. Temos ainda*

$$\hat{x} = (1, 0, 0, 0, 1, 1, 1, 1) \text{ e } \hat{x}' = (1, 0, 0, 1, 0, 0, 0, 0),$$

logo $\omega_H(\hat{x}) = 5$ *e* $\omega_H(\hat{x}') = 2$. *Assim,* $\omega_4(x) = \omega_4(\hat{x}) = 8 = \omega_H(\hat{x}) + (4-1) \cdot \frac{\omega_H(\hat{x}')}{2}$, *ou seja, a relação no Lema 4.1.17 é claramente satisfeita.*

Agora demonstramos a capacidade de correção de um código de b-símbolos em um alfabeto $\mathcal{A}$ qualquer.

Proposição 4.1.19. *Um código* $\mathcal{C}$ *pode corrigir até* t *erros de* b*-símbolos se, e somente se,* $d_b(\mathcal{C}) \geq 2t + 1$.

Demonstração. Suponha que o código $\mathcal{C}$ contenha duas palavras u e v com b-distância de no máximo $2t$ entre elas, isto é, $\pi_b(u)$ difere em no máximo $2t$ b-símbolos de $\pi_b(v)$. Seja w um vetor de b símbolos que coincida com $\pi_b(u)$ em todas as posições i tais que $\pi_b(u)_i = \pi_b(v)_i$ e nos t primeiros b-símbolos nos quais $\pi_b(u)$ difere de $\pi_b(v)$. Além disso, deixe w coincidir com $\pi_b(v)$ nas posições restantes (se $d_b(u, v) < t$, tome $w = \pi_b(u)$). Assim $d_H(\pi_b(u), w) \leq t$ e $d_H(\pi_b(v), w) \leq t$.

Agora, suponha que w é recebida junto com a informação de que no máximo t erros de b-símbolos ocorreram. Então u e v podem ter sido transmitidos (ou até mesmo outra palavra do código). Assim, não existe uma maneira de decidir qual palavra foi transmitida, logo existe uma falha ao corrigir até t erros de b-símbolos.

Por outro lado, suponha que $d_b(\mathcal{C}) \leq 2t + 1$ e uma palavra w for recebida junto com a informação de que um erro de peso no máximo t ocorreu. Se houvessem duas palavras u e v com distância no máximo t de w, então, pela desigualdade triangular, teríamos

$$d_b(u, v) \leq d_b(u, w) + d_b(w, v) \leq 2t,$$

contradizendo a hipótese. Logo, existe uma única palavra u com distância máxima t de w e, portanto, podemos deduzir que u foi transmitido. □

Na próxima seção, vamos estudar a b-distância de um código "intercalado". Notamos que uma extensão do Teorema 2.4.3 não é direta de derivar neste caso, ou seja, se c é uma palavra de um código linear cíclico $\mathcal{C}$, então os vetores $\hat{c}$ e $\hat{c}'$ não pertencem necessariamente ao código $\mathcal{C}$ e assim não é possível usar o Lema 4.1.17 para chegar a um limite na b-distância mínima de um código cíclico binário.

4.2 Construção de códigos por intercalação

O esquema de intercalação estudado em [3] gera códigos $\mathcal{C}$ que satisfazem $d_P(\mathcal{C}) = 2d_H(\mathcal{C})$. Vamos mostrar como essa construção pode ser generalizada para $b \geq 3$ arbitrário e, assim, então geramos códigos que satisfazem $d_b(\mathcal{C}) = b \cdot d_H(\mathcal{C})$. Este resultado também mostra que o limite superior na b-distância mínima indicada no Corolário 4.1.10 é mantida. A notação padrão de (n, M, d) será usado para denotar os parâmetros de um código binário de comprimento n, tamanho M e distância mínima de Hamming d.

Definição 4.2.1. *Dada uma coleção de b códigos $\mathcal{C}_0, \ldots, \mathcal{C}_{b-1}$, o* **código intercalado** $\mathcal{C}$ *é definido como segue:*

$$\begin{aligned} \mathcal{C} &= \{(c_{0,0}, \ldots, c_{b-1,0}, c_{0,1}, \ldots, c_{b-1,1}, \ldots, c_{0,n-1}, \ldots, c_{b-1,n-1}); \\ &\quad c_i = (c_{i,0}, \ldots, c_{i,n-1}) \in \mathcal{C}_i, \text{ para } 0 \leq i \leq b-1\}. \end{aligned} \tag{4.1}$$

Teorema 4.2.2. *Seja $\{\mathcal{C}_0, \ldots, \mathcal{C}_{b-1}\}$ um conjunto de b códigos binários com respectivos parâmetros (n, M_i, d_i), para $0 \leq i \leq b-1$. Então o código intercalado (Definição 4.2.1) é um $(bn, \prod_{i=0}^{b-1} M_i, \min_{0 \leq i \leq b-1}\{d_i\})$-código que satisfaz*

$$d_b(\mathcal{C}) = b \cdot d_H(\mathcal{C}) = b \cdot \min_{0 \leq i \leq b-1}\{d_i\}.$$

Demonstração. Claramente o comprimento de $\mathcal{C}$ é bn e $M = \prod_{i=0}^{b-1} M_i$, pois é o número de combinações possíveis das palavras de $\mathcal{C}_0, \ldots, \mathcal{C}_{b-1}$ como em (4.1). Note que se $u, v \in \mathcal{C}$, então

$$\begin{aligned} d_H(u, v) &= |\ \{(i, j) \mid u_{ij} = v_{ij},\ 0 \leq i \leq b-1,\ 0 \leq j \leq n-1 \text{ e } u_i, v_i \in \mathcal{C}\}\ | \\ &= \sum_{i=0}^{b-1} |\ \{j \mid u_{ij} = v_{ij},\ 0 \leq j \leq n-1\}\ | = \sum_{i=0}^{b-1} d_H(u_i, v_i), \end{aligned}$$

com $u_i, v_i \in \mathcal{C}_i$. Daí,

$$\begin{aligned} d_H(\mathcal{C}) &= \min\{d_H(u, v) \mid\ u, v \in \mathcal{C} \text{ e } u \neq v\} \\ &= \min\left\{\sum_{i=0}^{b-1} d_H(u_i, v_i) \mid u_i, v_i \in \mathcal{C}_i,\ 0 \leq i \leq b-1 \text{ e } u_i \neq v_i\right\} \\ &= \min\{d_i \mid\ 0 \leq i \leq b-1\}. \end{aligned}$$

De fato, se $c_i, u_i \in \mathcal{C}_i$ são tais que $d_H(c_i, u_i) = d_i$, para

$$d_H((0, \ldots, c_{i,0}, 0, \ldots, 0, c_{i,n-1}, 0, \ldots, 0), (0, \ldots, u_{i,0}, 0, \ldots, 0, u_{i,n-1}, 0, \ldots, 0))$$

$$= d_i \leq \sum_{i=0}^{b-1} d_H(u_i, v_i), \text{ para todos } i = 0, \ldots, b-1.$$

Logo, $d_H(\mathcal{C}) = \min\{d_i;\ 0 \leq i \leq b-1\}$.

Seja $c \in \mathcal{C}$. Então $c = (c_{0,0}, \ldots, c_{b-1,0}, c_{0,1}, \ldots, c_{b-1,1}, \ldots, c_{0,n-1}, \ldots, c_{b-1,n-1})$, obtido intercalando palavras $c_i = (c_{i,0}, \ldots, c_{i,n-1}) \in \mathcal{C}_i,\ 0 \leq i \leq b-1$.

Se $c \neq \mathbf{0}$, então $c_i \neq \mathbf{0}$, para algum $0 \leq i \leq b-1$. Os símbolos em c_i são separados por b posições de um para o outro em c e, assim, para cada coordenada j de c_i não nula, $\pi_b(c)_{j-l}$ é não nulo, com $0 \leq l \leq b-1$, isto é, existem no mínimo $b \cdot \omega_H(c_i)$ símbolos em $\pi_b(c)$ que não são nulos. Assim,

$$\omega_b(c) \geq b \cdot \omega_H(c_i) \geq b \cdot \min_{0 \leq i \leq b-1}\{d_i\} = b \cdot d_H(\mathcal{C}).$$

Em particular, $d_b(\mathcal{C}) = \omega_b(\mathcal{C}) \geq b \cdot d_H(\mathcal{C})$.

A inequação contrária segue do fato de que existe um palavra $c \in \mathcal{C}$ tal que $\omega_H(c) = d_H(\mathcal{C})$, como

$$\omega_H(c) \leq n \leq (b-1)(n-1) + n = bn - (b-1),$$

lembrando que bn é o comprimento de $\mathcal{C}$, podemos usar a Proposição 4.1.9,

$$d_b(\mathcal{C}) \leq \omega_b(c) \leq b \cdot \omega_H(c) = b \cdot d_H(\mathcal{C}).$$

Portanto, $d_b(\mathcal{C}) = b \cdot d_H(\mathcal{C})$, como afirmado.

□

De acordo com a Proposição 4.1.19 e com o Teorema 4.2.2, o código intercalado $\mathcal{C}$ pode corrigir até $\lfloor d_b(\mathcal{C}) - 1)/2 \rfloor$ erros de b-símbolos. Agora, descrevemos um algoritmo para $\mathcal{C}$ que alcança o raio de decodificação.

Um decodificador para códigos intercalados

Para um vetor $c = (c_0, \ldots, c_{n-1})$ definimos o vetor de comprimento bn

$$(c)_b = (c_0, \ldots, c_0, \ldots, c_{n-1}, \ldots, c_{n-1})$$

obtido repetindo b vezes cada coordenada em c. Agora, para cada componente $\mathcal{C}_i$, $0 \leq i \leq b-1$, do código intercalado $\mathcal{C}$, seja $(\mathcal{C}_i)_b$ o código de comprimento bn

$$(\mathcal{C}_i)_b = \{(c)_b;\ c \in \mathcal{C}_i\}.$$

Se $\mathcal{C}_i$ é um (n, M_i, d_i)-código, então $(\mathcal{C}_i)_b$ é um (bn, M_i, bd_i)-código. Assumimos que, para o código $\mathcal{C}_i$, exista um decodificador de distância limitada $\mathcal{D}_i$ que corrija erros de peso até o raio de decodificação. Como o código $(\mathcal{C}_i)_b$ pode ser interpretado com uma concatenação de um código externo $\mathcal{C}_i$ e um código interno de b repetições, como definido na Seção 1.9, podemos também assumir que $(\mathcal{C}_i)_b$ tem um decodificador $(\mathcal{D}_i)_b$ que pode corrigir até $\lfloor \frac{bd_i-1}{2} \rfloor$ erros. Para mais detalhes deste decodificador $(\mathcal{D}_i)_b$ a partir do decodificador $\mathcal{D}_i$, verificar a Seção 1.9 e o Exemplo 4.2.4.

Considere uma palavra $c \in \mathcal{C}$ dada por

$$c = (c_{0,0}, \ldots, c_{b-1,0}, c_{0,1}, \ldots, c_{b-1,1}, \ldots, c_{0,n-1}, \ldots, c_{b-1,n-1}),$$

com $c_i = (c_{i,0}, c_{i,1}, \ldots, c_{i,n-1}) \in \mathcal{C}_i$, para todo $0 \leq i \leq b-1$. Sejam $\pi_b(c)$ o vetor de b-símbolos de c e y o vetor de b-símbolos recebido de comprimento bn. Representamos y como

$$y = (y_0, \ldots, y_{bn-1}),$$

com $y_j = (y_{j,0}, \ldots, y_{j,b-1})$, para $0 \leq j \leq bn-1$. Assumimos

$$d_H(y, \pi_b(c)) \leq \lfloor (d_b(\mathcal{C}) - 1)/2 \rfloor.$$

Se indexarmos as posições das coordenadas em c de 0 a $bn-1$, a coordenada $c_{i,j}$ em c fica na posição $(jb+i)$, para $0 \leq i \leq b-1$, $0 \leq j \leq n-1$. Cada coordenada $c_{i,j}$ é lida b vezes, correspondendo aos componentes $y_{jb+i-(b-1),b-1}, \ldots, y_{jb+i-1,1}$, e $y_{jb+i,0}$, respectivamente.

Em seguida, combinamos essas estimativas de $c_{i,j}$ em um vetor binário $\overline{y}_{i,j}$ para $0 \leq i \leq b-1$, $0 \leq j \leq n-1$, no qual

$$\overline{y}_{i,j} = (y_{jb+i-(b-1),b-1}, \ldots, y_{jb+i-1,1}, y_{jb+i,0}).$$

Finalmente, o vetor y, tratado como um vetor binário, é particionado nos seguintes b vetores, cada um com comprimento bn:

$$\overline{y}_i = (\overline{y}_{i,0}, \overline{y}_{i,1}, \ldots, \overline{y}_{i,n-1}),$$

para $0 \leq i \leq b-1$.

Lema 4.2.3. *Para $0 \leq i \leq b-1$ e d_H a distância mínima de Hamming sobre o alfabeto binário $\mathcal{A}$, temos*

$$d_H(\overline{y}_i, (c_i)_b) \leq \lfloor (d_b(\mathcal{C}) - 1)/2 \rfloor.$$

Demonstração. Primeiro, para $0 \leq i \leq b-1$, todo $\overline{y}_i$ é uma versão com erros da palavra

$$(c_i)_b = (c_{i,0}, \ldots, c_{i,0}, \ldots, c_{i,n-1}, \ldots, c_{i,n-1}) \in (\mathcal{C}_i)_b.$$

Além disso, todo erro de b-símbolos em y pode mudar no máximo uma das entradas em todo $\overline{y}_i$ e assim a distância binária de Hamming entre $(c_i)_b$ e $\overline{y}_i$ é no máximo $\lfloor (d_b(\mathcal{C})-1)/2 \rfloor$, isto é,

$$d_H(\overline{y}_i, (c_i)_b) \leq \left\lfloor \frac{d_b(\mathcal{C})-1}{2} \right\rfloor.$$

□

Finalmente, do Teorema 4.2.2, temos

$$\left\lfloor \frac{d_b(\mathcal{C})-1}{2} \right\rfloor = \left\lfloor \frac{b.d_H(\mathcal{C})-1}{2} \right\rfloor \leq \left\lfloor \frac{b.d_i-1}{2} \right\rfloor,$$

para todo i, com $1 \leq i \leq b$. Assim, o decodificador $(\mathcal{D}_i)_b$ pode decodificar com sucesso a palavra $\overline{y}_i$. Então, para $0 \leq i \leq b-1$, a palavra c_i é decodificada com sucesso e, portanto, a palavra c também é.

Exemplo 4.2.4. *Considere os códigos:*

$$\begin{aligned} \mathcal{C}_0 &= \{0000000, 1101010\} \\ \mathcal{C}_1 &= \{0000000, 0111111\} \\ \mathcal{C}_2 &= \{0000000, 1001111, 1111001, 0110110\} \end{aligned}$$

com $d_H(\mathcal{C}_0) = 4$, $d_H(\mathcal{C}_1) = 6$ *e* $d_H(\mathcal{C}_2) = 4$. *Então o código intercalado é*

$$\begin{aligned} \mathcal{C} = \{&000000000000000000000, 00100000001001001001, 00100100100100000001, \\ &00000100100000100100 0, 00001001001001001001 0, 00101001001101101101 1, \\ &00101101101101001001 1, 00001101101001101101 0, 10010000010000010000 0, \\ &10110000010100110100 1, 10110100110100010000 1, 10010100110000110100 0, \\ &10011001011001011001 0, 10111001011101111101 1, 10111101111101011001 1, \\ &10011101111001111101 0\}. \end{aligned}$$

É facíl verificar que $d_H(\mathcal{C}) = 4$ *e* $d_3(\mathcal{C}) = 12 = 3 \cdot d_H(\mathcal{C})$, *como no Teorema 4.2.2. Suponha que seja recebida a palavra*

$$\begin{aligned} y = [&(111), (011), (100), (111), (110), (001), (011), (111), (111), (101), (001), (101), \\ &(010), (101), (011), (110), (100), (001), (011), (111), (111)]. \end{aligned}$$

Temos:

$$\overline{y}_{0,0} = (y_{0-2,2}, y_{0-1,1}, y_{0,0}) = (y_{n-2,2}, y_{n-1,1}, y_{0,0}) = (111)$$

$$\begin{array}{lllllllllllll}
\overline{y}_{0,1} & = & (y_{1,2}, y_{2,1}, y_{3,0}) & = & (101) & \overline{y}_{0,4} & = & (y_{10,2}, y_{11,1}, y_{12,0}) & = & (100) \\
\overline{y}_{0,2} & = & (y_{4,2}, y_{5,1}, y_{6,0}) & = & (000) & \overline{y}_{0,5} & = & (y_{13,2}, y_{14,1}, y_{15,0}) & = & (111) \\
\overline{y}_{0,3} & = & (y_{7,2}, y_{8,1}, y_{9,0}) & = & (111) & \overline{y}_{0,6} & = & (y_{16,2}, y_{17,1}, y_{18,0}) & = & (000).
\end{array}$$

Daí, $\overline{y}_0 = [(111), (101), (000), (111), (100), (111), (000)]$. *Pela Seção 1.9, para aplicar o decodificador* $(\mathcal{D}_i)_b$*, primeiramente, decodificamos o código interno de repetição de cada vetor* $\overline{y}_{0,i}$*, com* $0 \leq i \leq 6$*. Por exemplo, o vetor* $\overline{y}_{0,4}$ *possui duas entradas nulas e uma não nula, logo a decodificação de tal vetor é* 0*. Assim, obtemos* $c_0' = (1101010)$ *que pertence à* $\mathcal{C}_0$*, logo* $c_0 = c_0'$.

$$\overline{y}_{1,0} = (y_{-1,2}, y_{0,1}, y_{1,0}) = (y_{n-1,2}, y_{0,1}, y_{1,0}) = (110)$$

$$\begin{array}{lllllllllllll}
\overline{y}_{1,1} & = & (y_{2,2}, y_{3,1}, y_{4,0}) & = & (011) & \overline{y}_{1,4} & = & (y_{11,2}, y_{12,1}, y_{13,0}) & = & (111) \\
\overline{y}_{1,2} & = & (y_{5,2}, y_{6,1}, y_{7,0}) & = & (111) & \overline{y}_{1,5} & = & (y_{14,2}, y_{15,1}, y_{16,0}) & = & (111) \\
\overline{y}_{1,3} & = & (y_{8,2}, y_{9,1}, y_{10,0}) & = & (100) & \overline{y}_{1,6} & = & (y_{17,2}, y_{18,1}, y_{19,0}) & = & (111).
\end{array}$$

Daí, $\overline{y}_1 = [(110), (011), (111), (100), (111), (111), (111)]$. *Pela Seção 1.9, para aplicar o decodificador* $(\mathcal{D}_i)_b$*, primeiramente, decodificamos o código interno de repetição de cada vetor* $\overline{y}_{0,i}$*, com* $0 \leq i \leq 6$*. Assim, obtemos* $c_1' = (1110111)$ *que não pertence ao código. Como* $d_H(c_1', 0000000) = 6$ *e* $d_H(c_1', 0111111) = 2$*, temos* $c_1 = (0111111)$.

$$\begin{array}{lllll}
\overline{y}_{2,0} & = & (y_{0,2}, y_{1,1}, y_{2,0}) & = & (111) \\
\overline{y}_{2,1} & = & (y_{3,2}, y_{4,1}, y_{5,0}) & = & (110) \\
\overline{y}_{2,2} & = & (y_{6,2}, y_{7,1}, y_{8,0}) & = & (111) \\
\overline{y}_{2,3} & = & (y_{9,2}, y_{10,1}, y_{11,0}) & = & (101)
\end{array}
\qquad
\begin{array}{lllll}
\overline{y}_{2,4} & = & (y_{12,2}, y_{13,1}, y_{14,0}) & = & (000) \\
\overline{y}_{0,5} & = & (y_{15,2}, y_{16,1}, y_{17,0}) & = & (000) \\
\overline{y}_{2,6} & = & (y_{18,2}, y_{19,1}, y_{20,0}) & = & (111).
\end{array}$$

Daí, $\overline{y}_2 = [(111), (110), (111), (101), (000), (000), (111)]$. *Pela Seção 1.9, para aplicar o decodificador* $(\mathcal{D}_i)_b$*, primeiramente, decodificamos o código interno de repetição de cada vetor* $\overline{y}_{0,i}$*, com* $0 \leq i \leq 6$*. Assim, obtemos* $c_2' = (1111001)$ *que pertence à* $\mathcal{C}_2$*, logo* $c_2 = c_2'$.

Portanto, intercalando os vetores c_0*,* c_1 *e* c_2*, obtemos* $c = (10111101111111010110011)$.

4.3 Códigos com distância mínima de Hamming pequena

Nesta seção estudamos a b-distância mínima de duas classes especiais de códigos de b-símbolos. A primeira classe corresponde aos códigos "completos", isto é, quando consideramos $\mathcal{C} = \mathcal{A}^n$ (todo o espaço ambiente) e a segunda, quando consideramos $\mathcal{C}$ um código de Hamming linear e cíclico.

Dado um vetor x de comprimento n em $\mathcal{A}^n$, cada coordenada $0 \leq i \leq n$, está contida

em b coordenadas do vetor $\pi_b(x)$, a saber,

$$\begin{aligned}
\pi_b(x)_{i-b+1} &= (x_{i-b+1}, x_{i-b+2}, \ldots, x_i), \\
\pi_b(x)_{i-b+2} &= (x_{i-b+2}, x_{i-b+3}, \ldots, x_i, x_{i+1}), \\
&\vdots \\
\pi_b(x)_i &= (x_i, x_{i+1}, \ldots, x_{i+b-1}).
\end{aligned}$$

Seja $y = (y_0, y_1, \ldots, y_{n-1})$ um vetor recebido. O *decodificador majoritário* gera, para cada coordenada $0 \leq i \leq n$, o valor que mais se repete entre suas b coordenadas constituintes, ou ?, se b é par e o número de zeros e uns é igual. O lema seguinte fornece a b-distância mínima do código $\mathcal{C} = \mathcal{A}^n$ e prova que o decodificador majoritário pode ser usado para decodificar até a capacidade de correção do b-código de $\mathcal{C}$.

Lema 4.3.1. *Seja $\mathcal{C} = \mathcal{A}^n$. Para todo $b \geq 3$, a b-distância mínima satisfaz $d_b(\mathcal{C}) = b$ e o decodificador majoritário pode corrigir até $\lfloor \frac{b-1}{2} \rfloor$ b-símbolos de erros.*

Demonstração. Seja $x \in \mathcal{C}$ um vetor não nulo, se $0 < \omega_H(x) \leq n-(b-1)$, então $\omega_H(x) \geq 1$ e, pela Proposição 4.1.9,

$$\omega_b(x) \geq \omega_H(x) + b - 1 \geq 1 + b - 1 = b.$$

Se $\omega_H(x) > n - (b-1)$, isto é, $\omega_H(x) \geq n - (b-2)$. Vamos mostrar que $\omega_b(x) = n$. Seja $\omega_0(x) = \{i;\ 0 \leq i \leq n-1 \text{ e } x_i = 0\}$, então $\mid \omega_0(x) \mid = n - \omega_H(x) \leq n - (n-(b-2)) = b-2$, isto é, x possui no máximo $b-2$ entradas nulas. Como cada coordenada de $\pi_b(x)$ é uma sequência de b-símbolos, não há como uma dessas entradas ser nula. Assim, $\omega_b(x) = n > b$.

Portanto, para todo $x \in \mathcal{C}$ não nulo, $\omega_b(x) \geq b$.

Agora observe que, para qualquer palavra $c \in \mathcal{C}$, com $\omega_H(c) = 1$, existe $i = 0, \ldots, n-1$ tal que $c_i \neq 0$. Daí, $\pi_b(c)_i, \pi_b(c)_{i-1}, \ldots, \pi_b(c)_{i-(b-1)}$ são coordenadas não nulas de $\pi_b(c)$. Temos ainda $\pi_b(c)_j = \mathbf{0}$, para todo $j \in \{0, \ldots, n-1\} \setminus \{i-(b-1), i-(b-2), \ldots, i\}$, pois $c_j = 0$, para todo $j \neq i$, $j = 0, \ldots, n-1$. Logo, $\omega_b(x) = b$ e, portanto, $d_b(\mathcal{C}) = b$.

Suponha que existam $t \leq \lfloor \frac{b-1}{2} \rfloor$ b-símbolos com erros em uma versão recebida de $\pi_b(x)$. Então, para cada $i = 0, \ldots, n-1$, x_i pode estar errada em no máximo t dos b-símbolos aos quais x_i pertence, a saber, $\pi_b(x)_i, \pi_b(x)_{i-1}, \ldots, \pi_b(x)_{i-(b-1)}$. Como $t \leq \lfloor \frac{b-1}{2} \rfloor \leq \frac{b-1}{2} = \frac{b}{2} + \frac{1}{2} < \frac{b}{2}$, então em pelo menos $\frac{b}{2}+1$ dos $\pi_b(x)_j$, com $0 \leq j \leq n-1$, a coordenada x_i está correta (e são iguais). Logo o decodificador majoritário corrige esta como x_i e, portanto, corrige o vetor x. □

Fizemos um pequena alteração na demonstração do Lema 4.3.1, proposta por Yaakobi, Bruck and Siegel [28, Lemma 6], pois a afirmação $x \neq \mathbf{0}$ e $x \neq \mathbf{1}$, não implica em

$\omega_H(\hat{x}') \geq 2$, como é afirmado no artigo. De fato, se $x = (11011111)$, então $\hat{x} = (11111111)$ e $\hat{x}' = (00000000)$, assim $\omega_H(\hat{x}') = 0$.

Exemplo 4.3.2. *Seja* $\mathcal{C} = \{0,1\}^4$ *e* $b = 3$, *então*

$$\begin{aligned} C &= \{0000, 1000, 0100, 0010, 0001, 1100, 1010, 1001, 1110, 1011, 1101, 0110, \\ &\quad 0101, 0111, 0011, 1111\}. \end{aligned}$$

Daí $\omega_b(0000) = 0$, $\omega_b(1000) = \omega_b(0100) = \omega_b(0010) = \omega_b(0001) = 3$ *e* $\omega_b(1100) = \omega_b(1010) = \omega_b(1001) = \omega_b(1110) = \omega_b(1011) = \omega_b(1101) = \omega_b(0110) = \omega_b(0101) = \omega_b(0111) = \omega_b(0011) = \omega_b(1111) = 4$, *logo* $\omega_b(\mathcal{C}) = 3$.

Agora, suponha que recebemos o vetor $y = [(100), (001), (001), (010)]$. *Note que* $\pi_b(x) = [(x_0, x_1, x_2), (x_1, x_2, x_3), (x_2, x_3, x_0), (x_3, x_0, x_1)]$ *então a primeira coordenada de* y, (100) *corresponde à* (x_0, x_1, x_2), *a segunda* (001) *corresponde à* (x_1, x_2, x_3), *e assim sucessivamente. Daí, temos que* x_0 *vale 1,* x_1 *vale 0 e* x_2 *vale 0 em cada b-símbolo correspondente, logo* $x_0 = 1$ *e* $x_1 = x_2 = 0$. *Além disso,* x_3 *vale 1,0 e 0 em cada b-símbolo correspondente, assim o decodificador majoritário gera* $x_3 = 0$, *logo* $x = (1000)$.

O lema seguinte considera a b-distância de um código linear cíclico de Hamming, dando uma generalização da observação feita no Exemplo 2.3.6 de que códigos de pares de códigos de Hamming cíclicos, com $n = 2^m - 1$ e $m > 2$, possuem distância mínima de pares $d_P = 5$.

Lema 4.3.3. *Se* $\mathcal{C}$ *é um código de Hamming linear e cíclico, com comprimento* $n = 2^m - 1$ *e* $b + 2 \leq m$, *então* $d_b(\mathcal{C}) = 2b + 1$.

Demonstração. Seja $x \in \mathcal{C}$ uma palavra não nula. Para $\hat{x}$, definido na página 97, mostraremos que $\omega_b(\hat{x}) \geq 2b + 1$, daí como $\omega_b(\hat{x}) = \omega_b(x)$, temos $d_b(\mathcal{C}) \geq 2b + 1$. Note que, como $\hat{x}$ difere de x apenas nas sequências de no máximo $b - 2$ zeros consecutivos em x, temos $\omega_H(\hat{x}) \geq \omega_H(x) \geq d_H(\mathcal{C}) = 3$. Assuma $\hat{x} \neq \mathbf{1}$, então $\omega_H(\hat{x}')$ é um inteiro par positivo não nulo (Observação 2.4.2). Se $\omega_H(\hat{x}') \geq 4$, então, de acordo com o Lema 4.1.17, $\omega_b(x) \geq 3 + (b-1) \cdot \frac{4}{2} = 2b + 1$.

Se $\omega_H(\hat{x}') = 2$, então existem $i, j \in \{0, \ldots, n-1\}$, com $i \neq j$ tais que $\hat{x}'_i = 1$ e $\hat{x}'_j = 1$, e as demais coordenadas de $\hat{x}'$ todas nulas. Daí, $\hat{x}_i \neq \hat{x}_{i+1}$ e $\hat{x}_j \neq \hat{x}_{j+1}$. Note que não pode acontecer $\hat{x}_i = \hat{x}_j = 1$ e $\hat{x}_{i+1} = \hat{x}_{j+1} = 0$ e nem mesmo $\hat{x}_i \neq \hat{x}_j = 0$ e $\hat{x}_{i+1} = \hat{x}_{j+1} = 1$, pois, como $\hat{x}'_l = 0$, para todo $l \neq i, j$, no primeiro caso teríamos $\hat{x}_{i+l} = 0$, para todo $1 \leq l \leq j - i - 1$. Assim teríamos $\hat{x}'_{j-1} = \hat{x}_{j-1} + \hat{x}_j = 0 + 1 = 1$, contradizendo a hipótese. De forma análoga, verifica-se que não pode acontecer $\hat{x}_i \neq \hat{x}_j = 0$ e $\hat{x}_{i+1} = \hat{x}_{j+1} = 1$. Logo, ou $\hat{x}_i = \hat{x}_{j+1} = 1$ e $\hat{x}_{i+1} = \hat{x}_j = 0$ ou $\hat{x}_i = \hat{x}_{j+1} = 0$ e $\hat{x}_{i+1} = \hat{x}_j = 1$.

Se $\hat{x}_i = \hat{x}_{j+1} = 1$ e $\hat{x}_{i+1} = \hat{x}_j = 0$, então

$$\hat{x} = (\hat{x}_0, \ldots, \hat{x}_{i-1}, \hat{x}_i, \hat{x}_{i+1}, \ldots, \hat{x}_j, \hat{x}_{j+1}, \ldots, \hat{x}_{n-1}) = (1, \ldots, 1, 1, 0, \ldots, 0, 1 \ldots, 1).$$

Se $\hat{x}_i = \hat{x}_{j+1} = 0$ e $\hat{x}_{i+1} = \hat{x}_j = 1$, então

$$\hat{x} = (\hat{x}_0, \ldots, \hat{x}_{i-1}, \hat{x}_i, \hat{x}_{i+1}, \ldots, \hat{x}_j, \hat{x}_{j+1}, \ldots, \hat{x}_{n-1}) = (0, \ldots, 0, 0, 1, \ldots, 1, 0 \ldots, 0).$$

Portanto, $\hat{x}$ contém um única sequência de uns consecutivos, cujo comprimento s deve satisfazer $s \geq m$. Caso contrário, se $s < m$, as entradas não nulas da palavra x serão restritas a no máximo $m-1$. Se $g(x)$ é um polinômio gerador de grau m do código, isso significaria que existe um polinômio não nulo de grau no máximo $m-1$ que é um múltiplo de $g(x)$ (Corolário 1.4.9), o que é impossível. Assim, pelo Lema 4.1.17,

$$\omega_b(\hat{x}) \geq m + (b-1) \cdot \frac{2}{2} = m + b - 1 \geq 2b + 1.$$

Finalmente, notamos que se $\hat{x} = \mathbf{1}$, então $\omega_b(\hat{x}) = n \geq 2b + 1$, para $b \geq 3$ e $m \geq b + 2$.

Para mostrar que $d_b(\mathcal{C}) = 2b+1$, observe que $\mathcal{C}$ contém um palavra x tal que $\omega_H(x) = 3$, $x_0 = x_1 = 1$, $x_m = 1$ e, entre x_1 e x_m e entre x_m e x_0 há no mínimo $b-1$ coordenadas. Isso é possível, pois $f(x) = x^m + x + 1$ é irredutível sobre $\mathbb{F}_2$, daí $\mathbb{F}_{2^m} = \frac{\mathbb{F}_2[x]}{<x^m+x+1>}$ e existe α elemento primitivo de $\mathbb{F}_2^m$ tal que $\alpha^m + \alpha + 1 = 0$. Assim, como visto na página 17, $Hx^t = 1 + \alpha + \alpha^m = 0$, isto é, $x \in \mathcal{C}$. Além disso, x possui $m - 2 \geq b$ zeros entre x_1 e x_m e $2^m - 2 - m \geq b$ zeros entre x_m e x_0. Logo, $\omega_b(x) = 2b + 1$. □

Considerações Finais

Neste trabalho apresentamos o desenvolvimento inicial de duas novas classes de códigos, os *códigos para canais de leitura de pares de símbolos* e os *códigos para canais de leitura de b-símbolos*, fazendo um paralelo com a Teoria Clássica dos Códigos Corretores de Erros. Além das definições e resultados discutidos no decorrer deste trabalho, recentemente tomamos conhecimento de novas referências sobre o tema, as quais nos referimos sucintamente a seguir.

Em [11], Elisshco, Gabrysy e Yaakobi mostram como estender a cota de Johnson e a cota de programação linear para códigos de pares de símbolos e que estas novas cotas são melhores que as cotas apresentadas na Seção 2.5. Nesse artigo também são apresentadas construções de códigos com distância mínima relativa de pares pequena, a saber entre quatro e dez.

Sun, Zhu e Wang [26] estudaram a exata distância de pares de códigos cíclicos de comprimento p^ℓ sobre $\mathbb{F}^{p^m}$. Dinh *et al.* [9] estabeleceram a distância de pares de códigos λ-constacíclicos de comprimento p^s sobre $\mathbb{F}^{p^m}$, com p primo. Além disso, todos os códigos constacíclicos de pares de símbolos MDS de comprimento p^s são obtidos.

Chen, Lin e Liu [6] apresentaram três limitantes inferiores para a distância mínima de códigos constacíclicos, os dois primeiros são generalizações dos Teorema 2.3.5, Teorema 2.3.9 e [[16], Lemma 4.1]. O terceiro representa um limitante inferior para a distância mínima de códigos cíclicos com raízes repetidas. Além disso, eles fazem construções de novos códigos de pares MDS com distância mínima de pares sete e oito usando códigos cíclicos com raízes repetidas.

Dinh *et al.* [10] desenvolveram uma nova técnica para determinar a b-distância de todos os códigos constacíclicos de comprimento p^s sobre $\mathbb{F}_{p^m}$ para $1 \leq b \leq \left\lfloor \frac{p}{2} \right\rfloor$. Como uma aplicação desse método, também estabeleceram todos os códigos constacíclicos de b-símbolos MDS de comprimento p^s sobre o corpo finito $\mathbb{F}_{p^m}$.

Em [20], Mostafanasab e Sevim encontraram um novo método para calcular a b-distância entre dois vetores de comprimento n, com n natural. Eles também mostram

a b-distância de alguns códigos cíclicos de comprimento p^ℓ sobre $\mathbb{F}^{p^m}$.

Percebe-se desta maneira que as pesquisas sobre estes códigos ainda estão em andamento e bastante atuais pela demanda de tecnologias de transmissão de informações cada vez mais rápidas e eficientes.

Referências Bibliográficas

[1] R. E. Blahut, *Algebraic Codes on Lines, Plane and Curves, An Engineering Approach*. Cambridge University Press, 2008.

[2] D. H. Bueno-Carreño, J.J. Bernal, J.J. Simón, *Cyclic and BCH codes whose minimum distance equals their maximum BCH bound*, Advances in Mathematics of Communications, vol. 10, (2016) 459-474.

[3] Y. Cassuto, M. Blaum, *Codes for symbol-pair read channels*, IEEE Trans. Inf. Theory, vol. 57, no. 12, (2011) 8011-8020.

[4] Y. Cassuto, S. Litsyn, *Symbol-pair codes: Algebraic constructions and asymptotic bounds*, IEEE Trans. Inf. Theory, vol. 57, no. 12, (2011) 2348-2352.

[5] Y. M. Chee, L. Ji, H. M. Kiah, C. Wang, J. Yin, *Maximum distance separable codes for symbol-pair read channels*, IEEE Trans. Inf. Theory, vol. 59, no. 11, (2013) 7259-7267.

[6] B. Chen, L. Lin, H. Liu, *Constacyclic Symbol-Pair Codes: Lower Bounds and Optimal Constructions*, IEEE Trans. Inf. Theory, vol. 63, no. 12, Dec. (2017) 7661-7666.

[7] B. Ding, T. Zhang, G. Ge, *Maximum distance separable codes for b -symbol read channels*, Finite Field and Their Aplications, vol. 49, (2018) 180-197.

[8] B. Ding, G. Ge, T. Zhang, Y. Zhang, *New constructions of MDS Symbol-Pair codes*, Designs, Codes and Cryptography, vol. 86, no. 4, (2018) 841-859.

[9] H. Q. Dinh, B. T. Nguyen, A. K. Singh, S. Sriboonchitta, *On the Symbol-Pair Distance of Repeated-Root Constacyclic Codes of Prime Power Lengths*, IEEE Trans. Inf. Theory, vol. 64, no. 4, (2018) 2417-2430.

[10] H. Q. Dinh, X. Wang, H. Liu, S. Sriboonchitta, *On the b-Distance of Repeated-Root Constacyclic Codes of Prime Power Lengths**, Designs, Codes and Cryptography, vol. 86, no. 4, (2018) 841-859.

[11] O. Elishco, R. Gabrysy, E. Yaakobi, *Bounds and Constructions of Codes over Symbol-Pair Read Channels*, IEEE Trans. Inf. Theory, (2018) 2505-2509.

[12] C. Hartmann, K. Tzeng, *Generalizations of the BCH bound*, Information and Control, vol. 20, (1972) 489-498.

[13] A. Hefez, M.L.T. Villela, *Códigos Corretores de Erros*, 2^a Edição, Série Computação e Matemática **5**, IMPA, Rio de Janeiro, 2008.

[14] M. Hirotomo, M. Takita, M. Morii, *Syndrome decoding of symbol-pair codes*, in Proc. IEEE Trans. Inf. Theory, Hobart, TAS, Australia, (2014) 162-166.

[15] C. W. Huffman, V. Pless, *Fundamentals of Error-Correcting Codes*, Cambridge Univerty Press, New York, 2003.

[16] X. Kai, S. Zhu, P. Li, *A construction of new MDS symbol-pair codes*, IEEE Trans. Inf. Theory, Hobart, TAS, Australia, vol. 61, no. 11, (2015) 5828-5834.

[17] X. Kai , S. Zhu, Y. Zhao, H. Luo, Z. Chen, *New MDS Symbol-Pair Codes From Repeated-Root Codes*, IEEE Communications Letters, vol. 22, no. 3, March (2018) 462-465.

[18] S. Li, G. Ge, *Constructions of Maximum Distance Separable Symbol-Pair Codes Using Cyclic and Constacyclic Codes*, arXiv:1608.02687 (2016).

[19] F. MacWilliams, N. Sloane, *The Theory of Error-Correcting Codes*, The Netherlands: North Holland, Amsterdam, 1977.

[20] H. Mostafanasab, E. S. Sevim, *b-Symbol Distance Distribution of Repeated-Root Cyclic Codes*, arXiv:1704,03311 (2017).

[21] E. Orsini, M. Sala, *Correcting errors and erasures via the syndrome variety*, Jornal of Pure and Applied Algebra, vol. 200, (2005) 191-226.

[22] K. H. Rosen, *Discrete Mathematics and its Applications*, 6^a Edição, McGraw-Hill, New York, 2007.

[23] R. M. Roth, *Introduction to Coding Theory*, Cambridge University Press, Cambridge, U.K., 2005.

[24] H. Shahri, K. K. Tzeng, *On error-and-erasure decoding of cyclic codes*, IEEE Trans. Inf. Theory, vol. 38, no. 2, (1992) 489-496.

[25] C.E. Shannon, *A Mathematical Theory of communication*, The Bell System Technical Journal, vol. 27, no. 2, april (1948).

[26] Z. Sun, S. Zhu, L., Wang, *The symbol-pair distance distribution of a class of repeated-root cyclic codes over* $\mathbb{F}_{p^m}$, Cryptography and Communications, vol. 10, no. 4, July (2018) 643-653.

[27] J. H. van Lint, R. M. Wilson, *A Course in Combinatorics*, second edition, Cambridge Univ. Press, Cambridge, U.K., 2005.

[28] E. Yaakobi, J. Bruck, P. H. Siegel, *Constructions and decoding of cyclic codes over b-symbol read channels*, IEEE Trans. Inf. Theory, vol. 62, no. 4, (2016) 1541-1551.

[29] E. Yaakobi, J. Bruck, P. H. Siegel, *Decoding of cyclic codes over symbol-pair read channels*, in Proc. IEEE Trans. Inf. Theory, Cambridge, MA, USA, (2012) 2891-2895.

[30] H. Zhang, *Improvement on Minimum Distance of Symbol-Pair Codes*, Springer, IMACC 2017: Cryptography and Coding, Lecture Notes in Computer Science, vol 10655, Nov. (2017) 116-124.