DECODIFICAÇÃO DE CÓDIGOS ALGÉBRICO-GEOMÉTRICOS

Leocarlos B. S. Lima, Francisco M. Assis e Lirida A. B. Naviner

Resumo - O objetivo deste artigo é contextualizar o tema da decodificação de códigos algébrico-geométricos e estabelecer um panorama geral sobre as diferentes abordagens de decodificação para estes códigos. Além disso, são feitas considerações no que se refere ao desenvolvimento de arquiteturas de implementação eficiente em hardware para estes decodificadores, no sentido de tornar os códigos algébrico-geométricos competitivos em relação a códigos já estabelecidos, como os códigos de Reed-Solomon.

Palavras-chave: Códigos algébrico-geométricos, decodificação, implementação em hardware.

Abstract - The aim of this paper is to introduce the subject of decoding algebraic-geometric codes and to establish a general landscape about the different decoding approaches for these codes. Moreover, considerations are made regarding to the development of efficient hardware implementation architectures for these decoders, in order to make the algebraic-geometric codes competitive with regard to established codes, like Reed-Solomon codes.

Keywords: Algebraic-geometric codes, decoding, hardware implementation.

1. INTRODUÇÃO

Na codificação para controle de erros, os códigos de bloco mais amplamente utilizados têm sido os conhecidos códigos de Reed-Solomon (RS), que têm tido aplicação em diversas áreas, como no desenvolvimento de padrões para comunicação móvel celular, no armazenamento em CDs, na comunicação por satélite etc. [1, 2]. Apesar dessa larga utilização dos códigos RS, há atualmente outras opções na codificação de canal que apresentam melhores características assintóticas e começam a se mostrar competitivas, como é o caso dos *códigos algébrico-geométricos* (AG).

Os códigos AG nasceram da idéia do matemático russo V. D. Goppa, que na década de 70 sugeriu que, ao invés de avaliar polinômios em pontos simples (valores do corpo finito) como nos códigos RS, poder-se-ia avaliar funções algébricas em pontos de curvas definidas sobre corpos finitos [3]. Além disso, ele mostrou como substituir a condição imposta sobre graus de polinômios por condições sobre funções algébricas. Em 1982, M. A. Tsfasman, S. G. Vlăduţ e T. Zink combinaram a idéia de Goppa aos recentes resultados da geometria algébrica, produzindo os chamados códigos algébrico-geométricos ou códigos de Goppa geométricos [4–9]. Os códigos RS são um caso particular dos códigos AG quando a curva algébrica adotada é apenas uma reta. As curvas sobre corpos finitos podem ter muito mais pontos que uma simples reta, de forma que os códigos AG podem apresentar comprimento muito maior que os códigos RS.

Do ponto de vista da relação entre a taxa de informação assintótica, que representa o comprometimento da taxa de transmissão de informação devido à utilização do código (introdução de redundâncias), e a distância mínima relativa, que representa a capacidade de correção de erros do código, pode-se afirmar que os códigos AG fazem parte da classe dos chamados *bons códigos*, que são aqueles que não comprometem a taxa de transmissão em detrimento da capacidade de correção, ou vice versa, para um comprimento de código $n \to \infty$. Os códigos AG são, portanto, excelentes códigos que apresentam parâmetros como comprimento, capacidade de correção e taxa de informação melhores que códigos já estabelecidos, como os RS [6, 10]. Foi mostrado em 1982 por Tsfasman, Vlăduţ e Zink que códigos AG apresentam parâmetros que ultrapassam o limite de Gilbert-Varshamov, melhor limitante até então conhecido [4]. Recentemente, em 2001, C. Xing demonstrou que ambos os limites de Gilbert-Varshamov e de Tsfasman-Vlăduţ-Zink podem ser melhorados em torno dos pontos em que as curvas dos dois limitantes (figura 1) se interceptam [11].

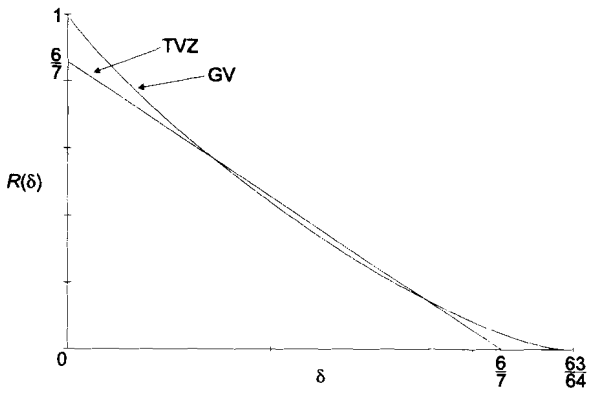

Figura 1. Comparação do limite de Gilbert-Varshamov (curva GV) com o limite de Tsfasman-Vlăduţ-Zink (curva TVZ) para $q = 64$.

O objetivo principal deste artigo é tratar do problema da decodificação dos códigos AG, descrevendo suas principais abordagens. É subentendido aqui um conhecimento prévio

Leocarlos B. S. Lima é doutorando pela Universidade Federal de Campina Grande (UFCG/COPELE), Campina Grande, PB, Brasil e pela École Nationale Supérieure des Télécommunications (ENST/COMELEC), Paris, France. Francisco M. Assis está ligado à UFCG/DEE, Campina Grande, PB, Brasil. Lirida A. B. Naviner está ligada à ENST/COMELEC, Paris, France.
E-mails: leocarlo@dee.ufcg.edu.br, leocarlos.lima@enst.fr, fmarcos@dee.ufcg.edu.br, lirida.naviner@enst.fr. Editor de Área responsável: Ricardo M. Campello de Souza. Artigo submetido em 20/Jul/2002, revisado em 06/Dez/2002, aceito em 29/Jan/2003.

de alguns conceitos da geometria algébrica, como ideais, variedades, curvas algébricas, corpos de funções, divisores, diferenciais, lacunas e anti-lacunas, bases de Gröbner etc. [12–18].

A seção que segue apresenta as definições dos códigos AG. Na seção 3, é tratado o problema da decodificação dos códigos AG e são apresentadas as abordagens de decodificação mais importantes destes códigos. A seção 4 trata do problema da implementação em hardware dos algoritmos de decodificação dos códigos AG. A seção 5 apresenta as conclusões.

2. CÓDIGOS ALGÉBRICO-GEOMÉTRICOS

São dois os diferentes tipos de construção de códigos AG: um chamado aqui de *construção por funções* e outro de *construção por diferenciais* [5, 16, 18–21].

Com relação à construção por funções, considere $\mathcal{X}$ uma curva projetiva não singular de gênero g, $P_1, \ldots, P_n$ pontos racionais de $\mathcal{X}$ e $D = P_1 + \cdots + P_n$ um divisor desta curva. Considere G um divisor de $\mathcal{X}$ cujo suporte seja disjunto de D, e suponha que $2g - 2 < \deg(G) < n$.

Definição 1 (Código por funções) *Define-se um código AG denotado por $C(D, G)$, sobre o corpo finito $\mathbb{F}_q$ de q elementos, como o mapeamento linear $\alpha : L(G) \rightarrow \mathbb{F}_q^n$ dado por*

$$\alpha(f) = (f(P_1), \ldots, f(P_n)). \tag{1}$$

em que $L(G) = \{f \in \mathbb{F}_q(\mathcal{X}) \mid (f) + G \succ 0\} \cup \{0\}$ é o espaço de funções gerado pelo divisor G, $\mathbb{F}_q(\mathcal{X})$ é o corpo de funções da curva $\mathcal{X}$, $(f) = \sum_{P_i} v_{P_i}(f) P_i$ é o divisor principal da função f e $v_{P_i}(f)$ é a ordem da função f no ponto $P_i \in \mathcal{X}$.

Do teorema de Riamann-Roch[1], mostra-se-se que a *dimensão* k de um código $C(D, G)$ é dada por

$$k = \deg(G) - g + 1. \tag{2}$$

Sua *distância mínima* d satisfaz a desigualdade $d \geq n - \deg(G)$, uma vez que qualquer função $f \in L(G)$ possui no máximo $\deg(G)$ zeros, ou seja, o peso de qualquer palavra código $\alpha(f)$ não é menor que $n - \deg(G)$. Do limite de Singleton[2], tem-se que

$$\begin{aligned} d \leq n - k + 1 = n - \deg(G) + g \Rightarrow \\ n - \deg(G) \leq d \leq n - \deg(G) + g. \end{aligned} \tag{3}$$

Com relação à construção de códigos AG por diferenciais, considere $\mathcal{X}$, $P_1, \ldots, P_n$, D e G da mesma forma anterior.

Definição 3 (Código por diferenciais) *Define-se um código AG denotado por $C^*(D, G)$, sobre $\mathbb{F}_q$ (corpo algebricamente fechado), como o mapeamento linear $\alpha^* : \Omega(G - D) \rightarrow \mathbb{F}_q^n$ dado por*

$$\alpha^*(\omega) = (\mathrm{Res}_{P_1}(\omega), \ldots, \mathrm{Res}_{P_n}(\omega)). \tag{4}$$

em que $\Omega(G - D) = \{\omega \in \Omega_{\mathcal{X}} \mid (\omega) \succ G - D\} \cup \{0\}$ é o espaço de diferenciais gerado pelo divisor $G - D$, $\Omega_{\mathcal{X}}$ é o conjunto de diferenciais associados à curva $\mathcal{X}$, $\mathrm{Res}_{P_i}(\omega)$ é o resíduo de ω no ponto P_i, $(\omega) = \sum_{P_i} v_{P_i}(f) P_i$ é o divisor do diferencial $\omega = f dt$ e $v_{P_i}(f) = v_{P_i}(\omega)$ é sua ordem no ponto $P_i \in \mathcal{X}$.

Considerando o teorema de Riemann-Roch (teorema 2), tem-se que a *dimensão* k^* do código $C^*(D, G)$ é dada por

$$k^* = n - \deg(G) + g - 1. \tag{5}$$

Sua *distância mínima* d^* satisfaz a desigualdade $d^* \geq \deg(G) - 2g + 2$ [22]. Do limite de Singleton, tem-se que

$$\begin{aligned} d^* \leq n - k^* + 1 = \deg(G) - g + 2 \Rightarrow \\ \deg(G) - 2g + 2 \leq d^* \leq \deg(G) - g + 2. \end{aligned} \tag{6}$$

Mostra-se que as duas construções $C(D, G)$ e $C^*(D, G)$ constituem códigos duais. Além disso, mostra-se que existe um diferencial ω com pólos simples e resíduo 1 nos pontos $P_1, \ldots, P_n$, tal que

$$C^*(D, G) = C(D, (\omega) + D - G). \tag{7}$$

em que (ω) é o divisor de ω [17, p. 48]. Isto implica que a construção por diferenciais fornece a mesma classe de códigos da construção por funções.

3. DECODIFICAÇÃO DE CÓDIGOS AG

O primeiro esquema de decodificação para os códigos AG foi proposto no final da década de 80 por Justesen et al [21] e consiste numa generalização do algoritmo PGZ (Peterson-Gorenstein-Zierler) [23] para decodificação de códigos BCH. Ele, de modo semelhante ao PGZ, fornece um polinômio localizador de erros que possui entre seus zeros as posições dos erros ocorridos na palavra recebida.

Em 1990, Skorobogatov e Vlăduţ [24] propuseram uma generalização do algoritmo de Justesen et al para curvas arbitrárias (ao invés de planas apenas). O algoritmo proposto ficou conhecido como *algoritmo básico* e é capaz de corrigir até $\frac{d-g-1}{2}$ erros ocorridos na palavra recebida, em que g é o gênero da curva usada na geração do código e d sua distância mínima projetada. Sua complexidade algorítmica é $\mathcal{O}(d^2 n + g^2 n) \leq \mathcal{O}(n^3)$, sendo n o comprimento do código. O algoritmo básico é hoje a base para uma grande quantidade de esquemas de decodificação de códigos AG encontrados na literatura. Ainda neste mesmo artigo, Skorobogatov e Vlăduţ apresentaram uma modificação do algoritmo básico, conhecida como *algoritmo modificado*, que corrige até $\frac{d-1}{2} - \sigma$ erros, em que σ é o chamado *defeito de Clifford* [24], que é aproximadamente igual a $\frac{g}{2}$ no caso de curvas planas. O algoritmo modificado tem complexidade $\mathcal{O}(n^4)$.

[1] Como referência, segue o importante resultado da geometria algébrica conhecido por teorema de Riemann-Roch.

Teorema 2 (Riemann-Roch) *Para um divisor G de uma curva de gênero g, tem-se que $l(G) = \deg(G) + 1 - g + i(G)$. Desta forma, se $\deg(G) > 2g - 2$, tem-se que $l(G) = \deg(G) + 1 - g$. Além disso, o índice de especialidade $i(G)$ (dimensão do espaço $\Omega(G)$) é dado por $i(G) = l(K - G)$ para todos os divisores G e divisores canônicos K.*

[2] O limite de Singleton afirma que, para um código linear (n, k, d) de comprimento n, dimensão k e distância mínima d, tem-se que $d \leq n - k + 1$.

Uma abordagem diferente na decodificação dos códigos AG, que consiste numa generalização do algoritmo de Euclides para solução da *equação chave* [25, 26], foi proposta por Porter em 1992 [27]. O algoritmo proposto corrige até $\frac{d-1}{2} - \sigma$ erros. Neste caso, o conjunto dos zeros correspondentes às posições dos erros é obtido de um ideal, ao invés de um polinômio. Mostrou-se que ambos, o algoritmo modificado e o algoritmo de Porter, são na verdade equivalentes. Esta formalização do problema da decodificação de códigos AG pela solução da equação chave tem sido usada também em outros decodificadores propostos [28–30].

O primeiro trabalho a proporcionar uma decodificação até metade da distância mínima foi proposto por Feng e Rao [31]. A elegante solução apresentada utiliza um esquema de decisão por maioria para determinar as síndromes desconhecidas da palavra recebida (apenas algumas síndromes são conhecidas e, uma vez conhecido um certo número de síndromes, pode-se determinar unicamente o vetor erro ocorrido). Este esquema de decisão por maioria de Feng e Rao foi aplicado posteriormente ao algoritmo de Porter por Shen e Tzeng [28].

As complexidades dos algoritmos acima descritos são proibitivas para as aplicações práticas, nas quais necessita-se utilizar códigos com comprimento elevado. No entanto, diversos esquemas rápidos de decodificação, ou seja, com menor complexidade, têm sido propostos. Em 1990, S. Sakata apresentou uma generalização em várias variáveis do clássico algoritmo de Berlekamp-Massey [23], que passou a ser conhecida como *algoritmo BMS* [32]. O algoritmo BMS, juntamente com o esquema de decisão por maioria de Feng e Rao, tem sido a base para a construção de diversos algoritmos de decodificação rápida para códigos AG [33–40].

Com o uso de matrizes de blocos de Hankel de códigos sobre curvas planas, Feng, Wei, Rao e Tzeng [41] obtiveram uma redução da complexidade do esquema de decisão por maioria. A implementação rápida proposta tem, no pior caso, complexidade $\mathcal{O}\left((r+1)n^2\right)$, em que $r+1$ é o grau da curva algébrica usada na definição dos códigos.

Em 1999, M. A. Shokrollahi e H. Wasserman apresentaram um esquema de decodificação de códigos AG chamado *decodificação de lista*, que consiste na busca pelo conjunto de palavras código que se encontram a uma determinada distância maior que a metade da distância mínima do código [42]. Esta idéia alternativa de decodificação foi proposta na década de 1950 por P. Elias [43] e J. M. Wozencraft [44], contrariando o conceito tradicional em que a decodificação consiste na busca pela única palavra código, caso exista, situada dentro da esfera de decodificação de diâmetro igual à distância mínima do código. Recentemente, alguns trabalhos têm sido publicados nesta linha envolvendo os códigos AG [45, 46].

Para maiores detalhes sobre a história da decodificação dos códigos AG, veja o artigo de Høholdt e Pellikaan [47].

No desenvolvimento de algoritmos de decodificação para códigos AG; observa-se de maneira geral algumas abordagens distintas:

1. Uma primeira abordagem, que é a utilizada no algoritmo básico de Skorobogatov e Vlăduţ [24], em que a síndrome é definida como um mapeamento de um subespaço linear de funções em um corpo de localização (corpo finito). Neste caso, a decodificação consiste na solução de um conjunto de equações lineares sobre este corpo;

2. Uma segunda abordagem, que é a utilizada no algoritmo de Porter [27], em que a síndrome é definida como um elemento em um anel afim. Neste caso, a decodificação consiste na solução de uma equação chave neste anel;

3. Uma terceira abordagem, da decodificação de lista [42], em que se admite a ocorrência de mais erros que a capacidade de correção do código;

4. Uma quarta abordagem que merece menção é a que envolve a utilização de esquemas de decisão suave, em que o demodulador fornece informações extras sobre a mensagem recebida ao decodificador, que as utiliza para melhorar seu desempenho na decodificação [36, 48–50].

De modo a ilustrar estas abordagens, serão descritos sucintamente nas subseções seguintes o algoritmo básico (primeira abordagem), o algoritmo de Porter (segunda abordagem), o algoritmo de Shokrollahi e Wasserman [42] (terceira abordagem), e o algoritmo GMD [51] (quarta abordagem). Além disso, serão vistos o esquema de decisão por maioria de Feng e Rao e o algoritmo BMS. Em seguida, será discutida a questão da implementação em hardware de decodificadores para códigos AG.

3.1 ALGORITMO BÁSICO (PRIMEIRA ABORDAGEM)

Considere $\mathcal{X}$ uma curva algébrica de gênero g. Considere $\mathcal{P}_{\mathcal{X}} = \{P_1, \ldots, P_n\}$ um conjunto de pontos racionais da curva $\mathcal{X}$ sob o corpo algebricamente fechado $\mathbb{F}_q$ e o divisor $D = P_1 + \cdots + P_n$, cujo suporte é $\mathcal{P}_{\mathcal{X}}$. Considere $G = aQ$ um divisor de grau $a \geq 0$, em que $Q \notin \mathcal{P}_{\mathcal{X}}$ é um ponto de $\mathcal{X}$ (suporte de G disjunto de $\mathcal{P}_{\mathcal{X}}$). Suponha também que $2g - 2 < a \leq n + g - 1$.

Considere nesta subseção o código $C^*(D, G)$ (dual do código $C(D, G)$)[3], que será denotado simplesmente por C, cuja matriz de paridade é $H_G = \|f_i(P_j)\|_{m \times n}$, em que $\{f_1, \ldots, f_m\}$ é uma base para o espaço $L(G)$.

Considere uma palavra recebida $\mathbf{v} \in \mathbb{F}_q^n$ e as funções $f_1, \ldots, f_m$ da base de $L(G)$. As síndromes de $\mathbf{v}$ são definidas por

$$S(\mathbf{v}, f_i) = \sum_{j=1}^{n} v_j f_i(P_j), \quad i = 1, \ldots, m, \tag{8}$$

caracterizando a primeira abordagem supracitada.

O algoritmo básico proposto por Skorobogatov e Vlăduţ [24] permite a correção simultânea de erros e apagamentos[4].

[3]Normalmente, os algoritmos de decodificação trabalham sobre códigos AG construídos por resíduos de diferenciais ($C^*(D, G)$), ao invés de códigos AG construídos pela avaliação de funções racionais ($C(D, G)$). Isto, porque no primeiro caso a descrição da matriz de paridade é mais simples, feita pela avaliação de funções racionais em pontos da curva.

[4]A única diferença entre erros e apagamentos é que as posições dos apagamentos são previamente conhecidas pelo decodificador, restando ao algoritmo determinar apenas os valores destes apagamentos.

Considere $\mathbf{e} \in \mathbb{F}_q^n$ o vetor de erros ocorridos, em que $w(\mathbf{e}) = t$, e $\mathbf{r} \in \mathbb{F}_q^n$ o vetor de apagamentos, sendo $w(\mathbf{r}) = \tau$. Denote por $\{E_1, \ldots, E_t\} \subset \mathcal{P}_\mathcal{X}$ e por $\{R_1, \ldots, R_\tau\} \subset \mathcal{P}_\mathcal{X}$ os conjuntos disjuntos de pontos de $\mathcal{P}_\mathcal{X}$ correspondentes às posições dos erros e dos apagamentos, respectivamente. Deve-se observar que, de fato, o algoritmo trata estas posições (pontos de $\mathcal{P}_\mathcal{X}$) como elementos de $\mathbb{F}_q$.

Além do divisor $G = aQ$, o algoritmo básico (algoritmo 4 a seguir) depende ainda de um divisor auxiliar $F = bQ$, em que $b \leq a$. A capacidade do algoritmo básico de corrigir t erros e τ apagamentos está condicionada a este divisor F, que pode ser determinado pelas inequações [24]

$$l(F) > t + \tau \tag{9}$$

e

$$a - b > t + 2g - 2. \tag{10}$$

Relativas a este divisor auxiliar F, são consideradas ainda as matrizes $H_F = \|k_i(P_j)\|_{s\times n}$, em que $\{k_1, \ldots, k_s\}$ é uma base para o espaço de funções $L(F)$, e $H_{G-F} = \|h_i(P_j)\|_{k\times n}$, sendo $\{h_1, \ldots, h_k\}$ uma base para o espaço $L(G - F)$.

Algoritmo 4 (Básico)

Entradas:

- *Uma palavra recebida* $\mathbf{v} = (v_1, \ldots, v_n)$;
- *O conjunto* $\{R_1, \ldots, R_\tau\}$ *das posições de apagamentos;*
- *As matrizes* H_G, H_F *e* H_{G-F}.

Saídas:

- *O vetor* $\mathbf{e} + \mathbf{r}$ *de erros e apagamentos.*

<<< **Passo 1** >>>

Determine a matriz $H_{F-R} = \|g_i(P_j)\|_{l\times n}$, *em que* $\{g_1, \ldots, g_l\}$ *é uma base para* $L(F - \sum R_i)$. *Observe que este espaço consiste nas funções de* $L(F)$ *que possuem zeros nas posições* $R_1, \ldots, R_\tau$. *Como* $L(F - \sum R_i) \subseteq L(F)$, *então as funções da base* $\{g_1, \ldots, g_l\}$ *podem ser escritas na forma* $\sum_j x_j k_j$, *sendo* $x_j \in \mathbb{F}_q$. *Tem-se, portanto, o sistema* $\sum_{j=1}^{s} k_j(R_i)x_j = 0$, *para* $1 \leq i \leq \tau$. *Do espaço de soluções deste sistema,* $l(F - \sum R_i) = \deg(F - \sum R_i) + 1 - g$ *soluções linearmente independentes podem ser obtidas. As combinações lineares das funções da base de* $L(F)$ *correspondentes às soluções obtidas neste sistema determinam as funções da base de* $L(F - \sum R_i)$, *das quais deriva a matriz* H_{F-R}.

<<< **Passo 2** >>>

Observe que $g_i h_j \in L(G)$. *Determine as* lk *síndromes* $S(\mathbf{v}, g_i h_j)$ *(equação 8), com* $i = 1, \ldots, l$ *e* $j = 1, \ldots, k$.

<<< **Passo 3** >>>

Determine uma solução não trivial $(y_1, \ldots, y_l)$ *para o sistema* $\sum_{j=1}^{l} S(\mathbf{v}, g_j h_i)\, x_j = 0$, *para* $1 \leq i \leq k$. *A condição da equação 9 garante que este sistema possui pelo menos uma solução não trivial.*

<<< **Passo 4** >>>

Dada a solução $(y_1, \ldots, y_l)$ *obtida no passo 3, determine o conjunto de zeros* $\{Q_1, \ldots, Q_u\} \subset \mathcal{P}_\mathcal{X}$ *da equação*

$$g_y = y_1 g_1 + \cdots + y_l g_l. \tag{11}$$

Isto é feito examinando-se os pontos de $\mathcal{P}_\mathcal{X}$ *um a um nesta equação 11. A condição da equação 10 garante que* g_y *possui entre seus zeros as posições dos erros e apagamentos ocorridos.*

<<< **Passo 5** >>>

Determine as síndromes $S(\mathbf{v}, f_i)$ *(equação 8), com* $i = 1, \ldots, m$.

<<< **Passo 6** >>>

Determine uma solução para o sistema $\sum_{j=1}^{u} f_i(Q_j)x_j = S(\mathbf{v}, f_i)$, *para* $1 \leq i \leq m$, *que determina os valores dos erros e apagamentos indicados por* g_y *(equação 11).*

Prova. Para detalhes sobre a prova deste algoritmo, veja [24]. ∎

O algoritmo básico possui uma capacidade de correção de t erros e τ apagamentos, em que

$$2t + \tau \leq d - g - 1 = a - 3g + 1. \tag{12}$$

Com relação à sua complexidade algorítmica, obedecido o limite acima para t e τ (equação 12), tem-se que ela não é maior que $\mathcal{O}(d^2 n + g^2 n) \leq \mathcal{O}(n^3)$ [24].

3.2 ALGORITMO BMS

A grande parte dos primeiros algoritmos de decodificação para códigos AG está baseada no processo de eliminação de Gauss, o que implica uma complexidade algorítmica $\mathcal{O}(n^3)$, em que n é o comprimento do código. Este nível de complexidade é proibitivo para aplicações práticas, em que são necessários códigos com comprimento elevado. Em vista da necessidade de esquemas de decodificação para códigos AG mais rápidos, diversos trabalhos vêm sendo apresentados neste sentido. O algoritmo BMS (Berlekamp-Massey-Sakata) [32, 37], tem dado origem a algoritmos rápidos de decodificação associado ao esquema de decisão por maioria proposto por Feng e Rao [31]. O primeiro é descrito nesta subseção e o segundo na subseção que segue.

Dado um inteiro $\mu > 0$, considere $\mathbb{Z}_+^\mu$ o conjunto das μ-úplas de inteiros não negativos. Sendo $\alpha = (\alpha_1, \ldots, \alpha_\mu) \in \mathbb{Z}_+^\mu$, considere a notação $x^\alpha = x_1^{\alpha_1} x_2^{\alpha_2} \ldots x_\mu^{\alpha_\mu}$ para um monômio nas μ variáveis $x_1, \ldots, x_\mu$. Defina $f(x_1, \ldots, x_\mu) = \sum_\alpha f_\alpha x^\alpha$ como sendo um polinômio em $\mathbb{F}_q[x_1, \ldots, x_\mu]$.

Denote por S um arranjo de dimensão μ de elementos $S_\alpha \in \mathbb{F}_q$, em que $\alpha \in \mathbb{Z}_+^\mu$. Diz-se que este arranjo S satisfaz a *relação de recursão linear μ-dimensional* com polinômio característico f se

$$\sum_\alpha f_\alpha S_{\alpha+\gamma} = 0, \tag{13}$$

para todo $\gamma \in \mathbb{Z}_+^\mu$, em que $S_{\alpha+\gamma}$ existe. A relação de recursão linear μ-dimensional representada pelo polinômio f é dita ser *válida para o arranjo* S se a equação 13 for satisfeita. O

conjunto dos polinômios característicos de todas as relações de recursão linear μ-dimensionais válidas para o arranjo S constitui um ideal de funções e será denotado aqui por $\mathbf{I}(S)$.

O algoritmo BMS proposto por Sakata constitue uma generalização em μ variáveis do algoritmo de Berlekamp-Massey para decodificação de códigos BCH [23]. O algoritmo de Berlekamp-Massey determina através de relações de recursão linear (registradores de deslocamento) em uma variável, relações estas válidas para as síndromes da palavra recebida, um polinômio localizador dos erros ocorridos. Já o algoritmo BMS, que tem como entrada um arranjo μ-dimensional S de elementos de $\mathbb{F}_q$, fornece um conjunto de polinômios mínimos característicos (uma base de Gröbner mínima para o ideal de funções $\mathbf{I}(S)$) correspondente às relações de recursão linear μ-dimensionais válidas para o arranjo S dado.

Num processo de decodificação, em que $\mathbf{v} = \mathbf{c} + \mathbf{e}$ é o vetor recebido, sendo $\mathbf{e}$ o vetor de erros ocorridos, se S é um arranjo μ-dimensional de síndromes de $\mathbf{v}$, então S satisfaz as relações de recursão linear μ-dimensionais com polinômio característico f se e só se $f(P) = 0$, para todo $P \in \sup(\mathbf{e})$. Portanto, o algoritmo BMS é utilizado para construir, a partir de um conjunto completo de síndromes de um vetor recebido, uma base para um ideal de funções (conjunto de polinômios mínimos característicos) em cujos zeros estão as posições dos erros ocorridos. No entanto, deve-se observar que o algoritmo BMS consiste apenas num dos componentes de um esquema de decodificação, uma vez que apenas uma parte do conjunto de síndromes é conhecido. Um algoritmo de decodificação necessita ainda de um componente adicional que determine as síndromes desconhecidas, tal como o esquema de decisão por maioria descrito na subseção 3.3.

3.2.1 ORDENAÇÃO DE MONÔMIOS

Considere agora a chamada ordenação de monômios lexicográfica graduada reversa. Nesta ordenação, que é denotada por $<_{grev}$, sendo $\alpha = (\alpha_1, \ldots, \alpha_\mu)$ e $\beta = (\beta_1, \ldots, \beta_\mu)$, diz-se que $x^\alpha <_{grev} x^\beta$ ou $\alpha <_{grev} \beta$ se e só se $|\alpha| = \sum_i \alpha_i < |\beta| = \sum_i \beta_i$ ou $|\alpha| = |\beta|$, $\alpha_1 = \beta_1, \ldots, \alpha_{j-1} = \beta_{j-1}$ e $\alpha_j > \beta_j$. Por exemplo, para $\mu = 2$, tem-se que $(0,0) <_{grev} (1,0) <_{grev} (0,1) <_{grev} (2,0) <_{grev} (1,1) <_{grev} (0,2) <_{grev} (3,0) <_{grev} (2,1) <_{grev} (1,2) <_{grev} \cdots$, ou $1 <_{grev} x <_{grev} y <_{grev} x^2 <_{grev} xy <_{grev} y^2 <_{grev} x^3 <_{grev} x^2y <_{grev} xy^2 <_{grev} y^3 <_{grev} \cdots$. Dado um polinômio $f(x_1, \ldots, x_\mu) = \sum_\alpha f_\alpha x^\alpha$, segundo esta ordenação $<_{grev}$, denota-se por $\deg(f)$ e $\mathrm{lc}(f)$ o expoente e o coeficiente do monômio líder (monômio de maior ordem), respectivamente.

Será considerada também uma *ordenação simples* de μ-úplas denotada por $<$, em que $\alpha < \beta$ se e só se $\alpha_i < \beta_i$, para todo $1 \leq i \leq \mu$. Esta ordenação simples não constitui propriamente uma ordenação de monômios, mas servirá para expressar a divisibilidade de um monômio por outro.

3.2.2 CONJUNTO DE POLINÔMIOS MÍNIMOS

Uma base de Gröbner mínima $\mathcal{F}$ para o ideal $\mathbf{I}(S)$ descrito há pouco consiste em polinômios característicos de relações de recursão linear μ-dimensionais que sejam válidas para o arranjo μ-dimensional S, e que possuam apenas monômios líderes mínimos segundo a ordenação de monômios $<_{grev}$ adotada. Diz-se, daí, que $\mathcal{F}$ é um *conjunto de polinômios mínimos* para o arranjo S de entrada.

Considere agora o caso em que a equação 13 é satisfeita apenas em parte do arranjo S. Considere os elementos válidos do arranjo S ordenados segundo a ordem $<_{grev}$ de seus índices (μ-úplas). Se S_α é o elemento válido de maior ordem, em que $\alpha = \deg(f) + \gamma$, para todo $\gamma \in \mathbb{Z}_+^\mu$, então a equação 13 pode ser escrita expressando S_α em função dos demais elementos S_β, em que $\beta <_{grev} \alpha$, ou seja,

$$S_\alpha = \frac{-1}{\mathrm{lc}(f)} \sum_{\beta <_{grev} \alpha} f_{\deg(f)-\alpha+\beta} S_\beta. \qquad (14)$$

A relação de recursão linear μ-dimensional representada pelo polinômio $f(x_1, \ldots, x_\mu)$ é dita agora *válida para o arranjo S até a entrada S_α*, se $\alpha \geq \deg(f)$ (ordenação simples) e a equação 14 for satisfeita, ou se $\alpha \ngeq \deg(f)$. Caso contrário, se $\alpha \geq \deg(f)$ e a equação 14 não for satisfeita, ela é dita *inválida*.

O conjunto dos polinômios característicos de todas as relações de recursão linear μ-dimensionais válidas para o arranjo S até a entrada S_α é denotado por $\mathbf{I}_\alpha(S)$. Observe que este conjunto $\mathbf{I}_\alpha(S)$ não constitui um ideal, uma vez que não é fechado sob a adição. Apesar disso, é fechado sob a multiplicação de monômios, ou seja, se $f(x_1, \ldots, x_\mu) \in \mathbf{I}_\alpha(S)$, então $x^\gamma f(x_1, \ldots, x_\mu) \in \mathbf{I}_\alpha(S)$. Além disso, a definição de um conjunto de polinômios mínimos para este conjunto $\mathbf{I}_\alpha(S)$ coincide ainda com a definição de uma base de Gröbner.

Reformulando a equação 14, tem-se que um polinômio f pertence ao conjunto $\mathbf{I}_\alpha(S)$ se e só se

$$\sum_\alpha f_\alpha S_{\alpha+\gamma} = 0, \qquad (15)$$

para todo $\gamma \in \mathbb{Z}_+^\mu$, tal que $\deg(f) + \gamma \leq_{grev} \alpha$.

3.2.3 CONJUNTO DE SENTINELAS E CONJUNTO DELTA

Defina *conjunto delta*, denotado por $\Delta(\mathbf{I}_\alpha(S))$, como o conjunto dos monômios (na verdade, índices) que não constituem termos líderes em qualquer polinômio de $\mathbf{I}_\alpha(S)$. Por esta razão, Sakata denominou $\Delta(\mathbf{I}_\alpha(S))$ de *conjunto de pontos excluídos*. Um conjunto $\mathcal{F} \subset \mathbf{I}_\alpha(S)$ é um conjunto de polinômios mínimos para $\mathbf{I}_\alpha(S)$ se $\Delta(\mathcal{F}) = \Delta(\mathbf{I}_\alpha(S))$.

A validade dos polinômios característicos de um conjunto $\mathcal{F}$, que constitui a saída do algoritmo BMS, pode ser verificada pela equação 15, contudo é necessário ainda verificar se os polinômios de $\mathcal{F}$ são mínimos (se seus monômios líderes são mínimos). Para realizar esta verificação, será necessário utilizar um segundo conjunto $\mathcal{G}$ de polinômios, chamado *conjunto de sentinelas*, definido em seguida.

Considere f um polinômio característico de uma relação de recursão linear μ-dimensional válida para o arranjo S até todas as entradas S_β, com $\beta <_{grev} \alpha$, mas não necessariamente até S_α. Defina o *valor predito* P_α para a entrada S_α

associado ao polinômio f pela equação

$$P_\alpha(f) = \frac{-1}{\mathrm{lc}(f)} \sum_{\beta <_{grev} \alpha} f_{\deg(f)-\alpha+\beta} S_\beta. \quad (16)$$

Observe que esta expressão consiste simplesmente no lado direito da equação 14. Diz-se, então, que a relação de recursão linear μ-dimensional representada pelo polinômio f é válida até a entrada S_α do arranjo S se e só se o valor real de S_α for igual ao valor predito P_α.

Considere agora um polinômio característico $g(x_1, \ldots, x_\mu)$ de uma relação de recursão linear μ-dimensional que é válida para o arranjo S μ-dimensional até todas as entradas S_β, com $\beta <_{grev} \alpha$, mas que é inválida até a entrada S_α. Defina, então, a *extensão* de g como sendo o vetor (índice) dado por

$$\mathrm{Span}(g) = \alpha - \deg(g) \quad (17)$$

e sua *discrepância* por

$$\delta_g = \mathrm{lc}(g)\,[S_\alpha - P_\alpha(g)] = \sum_\alpha g_\alpha S_{\alpha+\mathrm{Span}(g)} \neq 0. \quad (18)$$

Caso $g \notin \mathbf{I}_\alpha(S)$, ou seja, caso $P_\alpha(g) \neq S_\alpha$, tem-se que $\mathrm{Span}(g) \in \Delta(\mathbf{I}_\alpha(S))$. Neste caso, o polinômio $g(x_1, \ldots, x_\mu)$ é dito ser um *sentinela* para o ponto $\mathrm{Span}(g)$.

Defina um conjunto Δ como tendo um canto interior (veja exemplo na tabela 1) associado a cada sentinela do conjunto de sentinelas $\mathcal{G} \subset \mathbb{F}_q[x_1, \ldots, x_\mu] \backslash \mathbf{I}(S)$. Estes cantos interiores de Δ são também membros do conjunto $\Delta(\mathbf{I}(S))$.

	0	1	2	3
0	•	○	•	•
1	◇	•	•	•
2	○	•	•	•
3	•	•	•	•

Tabela 1. Ilustração de um conjunto $\Delta = \{(1,0)\}$. Os pontos externos de Δ estão representados por símbolos ○ e os pontos internos por símbolos ◇. Observe que estes pontos constituem realmente cantos na tabela.

Por fim, a verificação de se um conjunto $\mathcal{F}$ é um conjunto de polinômios mínimos, dado um conjunto de sentinelas $\mathcal{G}$, é feita com base no fato que segue. Se $\mathcal{F} \subset \mathbf{I}_\alpha(S)$ e $\mathcal{G} \subset \mathbb{F}_q[x_1, \ldots, x_\mu] \backslash \mathbf{I}_\alpha(S)$ é um conjunto de sentinelas para o conjunto delta $\Delta(\mathcal{F})$, então $\Delta(\mathcal{F}) = \Delta(\mathbf{I}_\alpha(S))$, o que implica que $\mathcal{F}$ é um conjunto de polinômios mínimos para $\mathbf{I}_\alpha(S)$. Se $\mathcal{F} \subset \mathbf{I}(S)$ e $\mathcal{G} \subset \mathbb{F}_q[x_1, \ldots, x_\mu] \backslash \mathbf{I}(S)$ é um conjunto de sentinelas para o conjunto delta $\Delta(\mathcal{F})$, então $\Delta(\mathcal{F}) = \Delta(\mathbf{I}(S))$, o que implica que $\mathcal{F}$ é uma base de Gröbner mínima para o ideal $\mathbf{I}(S)$.

3.2.4 DESCRIÇÃO DO ALGORITMO

O algoritmo BMS (algoritmo 5) basicamente opera sobre dois conjuntos: um conjunto de polinômios mínimos $\mathcal{F}$ e um conjunto de sentinelas $\mathcal{G}$. Cada iteração do algoritmo toma como entrada um conjunto de polinômios mínimos $\mathcal{F}$ para um conjunto $\mathbf{I}_\alpha(S)$ e um conjunto de sentinelas $\mathcal{G}$ para um conjunto delta $\Delta = \Delta(\mathbf{I}_\alpha(S))$, e produz um conjunto de polinômios mínimos $\mathcal{F}^+$ para um conjunto $\mathbf{I}_{\alpha^+}(S)$ e conjunto de sentinelas $\mathcal{G}^+$ para um conjunto delta Δ^+ $\Delta(\mathbf{I}_{\alpha^+}(S))$, sendo α^+ o índice de ordem imediatamente perior a α (ordenamento $<_{grev}$).

Algoritmo 5 (Algoritmo BMS)

Entradas:

- *Um arranjo μ-dimensional S de elementos de $\mathbb{F}_q$;*
- *Um índice $\alpha \in \mathbb{Z}_+^\mu$;*
- *Um conjunto de polinômios mínimos $\mathcal{F}$ para $\mathbf{I}_\alpha(S)$;*
- *Um conjunto de sentinelas $\mathcal{G}$ para $\Delta(\mathbf{I}_\alpha(S))$, com suas extensões e discrepâncias.*

Saídas:

- *Um conjunto de polinômios mínimos $\mathcal{F}^+$ para $\mathbf{I}_{\alpha^+}(S)$,*
- *Um conjunto de sentinelas $\mathcal{G}^+$ para $\Delta(\mathbf{I}_{\alpha^+}(S))$, com suas extensões e discrepâncias.*

<<< **Passo 1** >>>

Considere o conjunto $\mathcal{F}' = \{f \in \mathcal{F} \mid \deg(f) \leq \alpha^+\}$. Para cada $f \in \mathcal{F}'$, calcule o valor predito $P_{\alpha^+}(f)$ (equação 16). Considere o conjunto $\mathcal{N} = \{f \in \mathcal{F}' \mid P_{\alpha^+}(f) \neq S_{\alpha^+}\}$.

<<< **Passo 2** >>>

Faça $\mathcal{G}^+ = \mathcal{G} \cup \mathcal{N}$. Para cada $f \in \mathcal{N}$, calcule e armazene a extensão $\mathrm{Span}(f)$ (equação 17). Faça $\Delta^+ = \Delta \cup \{\mathrm{Span}(f) \mid f \in \mathcal{N}\}$. Para cada $f \in \mathcal{N}$, calcule e armazene a discrepância δ_f (equação 18).

<<< **Passo 3** >>>

Para cada $\beta \in \mathrm{Ext}\,\Delta^+$ (cantos externos de Δ^+), proceda o seguinte:

- *Primeiro, se existir um $f \in \mathcal{F} \backslash \mathcal{N}$, tal que $\deg(f) = \beta$, então faça $h^{(\beta)}(x_1, \ldots, x_\mu) = f(x_1, \ldots, x_\mu)$;*
  - *Em caso contrário, se $\beta \not\leq \alpha^+$, tome um $f \in \mathcal{N}$, tal que $\deg(f) \leq \beta$, e faça $h^{(\beta)}(x_1, \ldots, x_\mu) = x^{\beta-\deg(f)} f(x_1, \ldots, x_\mu)$;*
  - *Em caso contrário, tome um $g \in \mathcal{G}$, tal que $\mathrm{Span}(g) \geq \alpha^+ - \beta$, e um $f \in \mathcal{N}$, tal que $\deg(f) \leq \beta$. Considere os índices $q = \beta - \deg(f)$ e $p = \mathrm{Span}(g) - \alpha^+ + \beta$. Faça, então, $h^{(\beta)}(x_1, \ldots, x_\mu) = x^q f(x_1, \ldots, x_\mu) - \frac{\delta_f}{\delta_g} x^p g(x_1, \ldots, x_\mu)$;*

Por fim, tem-se que $\mathcal{F}^+ = \{h^{(\beta)}(x_1, \ldots, x_\mu) \mid \beta \in \mathrm{Ext}\,\Delta^+\}$.

Prova. Para detalhes sobre a prova deste algoritmo, veja [32,37]. ■

No passo 1 do algoritmo BMS (algoritmo 5), a validade dos polinômios de $\mathcal{F}$, que por hipótese correspondem a relações de recursão linear μ-dimensionais válidas para todas as entradas do arranjo S até a entrada S_α, é testada para a próxima entrada S_{α^+}. Os polinômios inválidos para a entrada S_{α^+} podem ser usados como sentinelas, sendo armazenados no conjunto $\mathcal{N}$. No passo 2, o conjunto de pontos excluídos

$\Delta = \Delta(\mathbf{I}_\alpha(S))$ é atualizado usando os novos sentinelas contidos no conjunto $\mathcal{N}$. Observe que pode ocorrer de um ou mais $f \in \mathcal{N}$ serem sentinelas de pontos excluídos $\mathrm{Span}(f)$ que já pertençam ao conjunto Δ. Além disso, deve-se levar em consideração que ao acrescentar um novo ponto excluído γ ao conjunto Δ, deve-se certificar que todos os pontos $\beta \leq \gamma$ também pertençam ao conjunto atualizado Δ^+ (acrescentá-los se necessário), de modo que este conjunto Δ^+ também seja um conjunto delta, segundo a definição. Os valores de $\mathrm{Span}(f)$ e δ_f são armazenados para um possível uso posteriormente no passo 3. O passo 3 consiste na determinação do conjunto atualizado $\mathcal{F}^+$ de polinômios válidos para o arranjo S até a entrada S_{α^+}. Este conjunto $\mathcal{F}^+$ é um conjunto de polinômios mínimos para $\mathbf{I}_{\alpha^+}(S)$ e $\mathcal{G}^+$ é um conjunto de sentinelas para $\Delta(\mathbf{I}_{\alpha^+}(S))$.

3.3 ESQUEMA DE DECISÃO POR MAIORIA

No corpo $\mathbb{C}$ dos números complexos, a transformada de Fourier discreta de um vetor $\mathbf{u} = (u_0, \ldots, u_{n-1})$ de números complexos é um vetor $\mathbf{U} = (U_0, \ldots, U_{n-1})$, em que $U_k = \sum_{i=0}^{n-1} e^{-j\frac{2\pi}{n}ki} u_i$, sendo $k = 0, \ldots, n-1$. Observe que o termo $e^{-j\frac{2\pi}{n}}$ constitui uma raiz n-ésima da unidade em $\mathbb{C}$, ou seja, $\left(e^{-j\frac{2\pi}{n}}\right)^n = 1$. Num corpo finito $\mathbb{F}_q$, um elemento α de ordem n também constitui uma raiz n-ésima da unidade. Por analogia, define-se a transformada de Fourier de um vetor $\mathbf{v} = (v_1, \ldots, v_n)$ em um corpo finito $\mathbb{F}_q$ como sendo o vetor $\mathbf{V} = (V_1, \ldots, V_n)$, em que

$$V_j = \sum_{i=1}^{n} \alpha^{ij} v_i, \; j = 1, \ldots, n, \tag{19}$$

sendo $\alpha \in \mathbb{F}_q$ um elemento de ordem n. O vetor $\mathbf{v}$ pode ser obtido pela transformada inversa dada por $v_i = \frac{1}{n}\sum_{j=1}^{n} \alpha^{-ij} V_j$, para $i = 1, \ldots, n$, em que $n \equiv 0 \bmod p$, sendo p um inteiro primo e $q = p^m$, para algum inteiro m.

Observe que a equação 19 equivale à expressão das síndromes de uma palavra recebida $\mathbf{v}$ para um código em uma variável (códigos BCH, RS, etc.). De um modo geral, a transformada de Fourier de um vetor recebido $\mathbf{v} = \mathbf{c} + \mathbf{e}$, em que $\mathbf{e}$ é um vetor erro, para um código (n, k) qualquer, pode ser vista como o conjunto das síndromes $S_i(\mathbf{v}) = S_i(\mathbf{e}) = \mathbf{h}_i \mathbf{e}^T$, para $i = 1, \ldots, n$, em que $\mathbf{h}_i$ são as linhas da matriz de paridade do código. O importante neste fato é que, uma vez conhecidas as n síndromes de $\mathbf{v}$ (apenas $n-k$ síndromes são conhecidas a princípio), é possível determinar o vetor $\mathbf{e}$ ocorrido através da transformada inversa de Fourier. O esquema de decisão por maioria proposto por Feng e Rao [31] permite se obter recursivamente as síndromes desconhecidas, possibilitando assim a decodificação de $\mathbf{v}$.

3.3.1 ANTI-LACUNAS

Considere uma curva algébrica plana $\mathcal{X}$ de gênero g sobre um corpo $\mathbb{F}_q$. Considere o divisor $D = P_1 + \cdots + P_n$ de pontos racionais de $\mathcal{X}$ e o ponto racional Q também de $\mathcal{X}$, mas disjunto do suporte de D. Sendo m um inteiro não negativo, considere divisores do tipo mQ (divisores de um único ponto).

Definição 6 (Lacuna) *Um inteiro não negativo m é dito ser uma* lacuna *("gap") de um ponto Q de uma curva $\mathcal{X}$, se $l(mQ) = l((m-1)Q)$.*

Pelo teorema de Riemann-Roch (teorema 2), valores de $m > 2g-1$ não constituem lacunas. Para valores de m até $2g-1$, observa-se que $1 = l(0) \leq l(Q) \leq \cdots \leq l((2m-1)Q) = g$. Ou seja, existem g valores diferentes de $l(iQ)$, para $0 \leq i \leq 2g-1$, e, portanto, um total de g lacunas.

Definição 7 (Anti-lacuna) *Um inteiro positivo m é dito ser uma* anti-lacuna *("nongap") de um ponto Q de uma curva $\mathcal{X}$, se $l(mQ) \neq l((m-1)Q)$, ou seja, se e só se existir uma função racional $f \in L(mQ)$, tal que $v_Q(f) = -m$ (possui pólos de ordem m em Q).*

Se m_i é uma anti-lacuna e $m_{i-1} < m_i$, então $0 = m_0 < m_1 < \cdots < m_{g-1} < m_g = 2g$ e $m_i = i + g$, para $i \geq g$. Também, se m_1 e m_2 são anti-lacunas de Q, então $m_1 + m_2$ também é uma anti-lacuna de Q. Portanto, as anti-lacunas de um ponto Q formam um semi-grupo na adição.

Denote por C_{m_k} o código AG $C^*(D, m_k Q)$ de comprimento n e dimensão $n - m_k + g - 1$. Assuma $m_k > 2g$ e considere $k = m_k - g + 1$ a dimensão do código dual. Denote por $(m_i \mid i \in \mathbb{N})$ a seqüência das anti-lacunas de Q, em que $0 < m_1 < \cdots < m_{g-1} < 2g$ e $m_i = i + g$, para $i = g, g+1, \ldots, m_k - g$. Denote por g_i uma função racional que possui um pólo de ordem m_i no ponto Q, e nenhum outro pólo mais. Tem-se que $g_1, \ldots, g_k$ constitui uma base para o espaço $L(m_k Q)$ ortogonal ao código C_{m_k}.

3.3.2 MATRIZ DE SÍNDROMES BI-DIMENSIONAIS

Se $\mathbf{e}$ é um vetor erro ocorrido em uma palavra recebida, defina $S = \|S_{i,j}\|_{k \times k}$ como a *matriz de síndromes bi-dimensionais* correspondentes a $\mathbf{e}$, cujos elementos são dados por $S_{i,j}(\mathbf{e}) = \sum_{l=1}^{t} e_{u_l} g_i(P_{u_l}) g_j(P_{u_l})$, em que os u_ls são as posições dos t erros ocorridos. Se $\mathbf{v} = \mathbf{c} + \mathbf{e}$ é uma palavra recebida, para $\mathbf{c} \in C_{m_k}$, e $m_i + m_j = m_p \leq m_k$, então $g_i g_j \in L(m_p Q) \subseteq L(m_k Q)$ e $S_{i,j}(\mathbf{e}) = S_{i,j}(\mathbf{v})$. Portanto, $S_{i,j}$ é uma entrada conhecida da matriz S se $m_i + m_j \leq m_k$ (conserve esta notação).

Defina o conjunto de pares N_k como sendo $N_k = \{(i,j) \in \mathbb{N}^2 \mid m_i + m_j = m_{k+1}\}$ e denote por n_k seu número de elementos. As entradas da matriz S com índices $(i,j) \in N_k$ são as primeiras síndromes desconhecidas com relação ao código C_{m_k} a serem determinadas. Uma vez determinada uma entrada $S_{i,j}$, com $(i,j) \in N_k$, então todas as outras entradas $S_{i',j'}$, com $(i',j') \in N_k$, podem ser obtidas. Isto, porque cada uma das funções $g_i g_j$, $g_{i'} g_{j'}$ e g_{k+1} gera o espaço $L(m_{k+1}Q) \setminus L(m_k Q)$. Ou seja, existem elementos $\lambda_{i,j}, \lambda_{i,j,r} \in \mathbb{F}_q$, com $\lambda_{i,j} \neq 0$, tais que $g_i g_j = \lambda_{i,j} g_{k+1} + \sum_{r \leq k} \lambda_{i,j,r} g_r \Rightarrow S_{i,j} = \lambda_{i,j} S_{k+1} + \sum_{r \leq k} \lambda_{i,j,r} S_r$, para todo $(i,j) \in N_k$. Além disso, esta relação é a mesma para qualquer vetor erro.

Considere agora a matriz $S(i,j) = \|S_{r,s}\|_{i \times j}$. Se $m_i + m_j = m_{k+1}$, então todos os elementos de $S(i,j)$ são conhecidos, exceto $S_{i,j}$. Se $m_i + m_j = m_k$, então $S(i,j)$ equivale à matriz de síndromes determinada no passo 2 do algoritmo básico (algoritmo 4) para o código C_{m_k} (m_k é o parâmetro a naquele algoritmo).

3.3.3 CANDIDATOS E DISCREPÂNCIAS

Considere $(i,j) \in N_k$, ou seja, $m_i + m_j = m_{k+1}$.

Definição 8 (Candidato) *Diz-se que um ponto (i,j) é um* candidato *com relação ao código C_{m_k} se as matrizes $S(i,j-1)$, $S(i-1,j-1)$ e $S(i-1,j)$ apresentam o mesmo posto.*

Se (i,j) é um candidato, então existe um único valor $S'_{i,j}$ a ser atribuído à entrada desconhecida $S_{i,j}$, de modo que as matrizes $S(i-1,j-1)$ e $S(i,j)$ possuam o mesmo posto. O elemento $S'_{i,j}$ é dito ser um *valor candidato* ou *predito* da síndrome desconhecida $S_{i,j}$.

Denote o número de candidatos *verdadeiros* ou *corretos*, em que $S'_{i,j} = S_{i,j}$, por T e o número de candidatos *falsos* ou *incorretos*, em que $S'_{i,j} \neq S_{i,j}$, por F.

Definição 9 (Discrepância) *Um índice (i,j) é dito ser uma* discrepância *(não confundir com o conceito de discrepância usado na descrição do algoritmo BMS na subseção 3.2) se as matrizes $S(i-1,j-1)$, $S(i-1,j)$ e $S(i,j-1)$ possuem um mesmo posto, que difere do posto de $S(i,j)$.*

Em se aplicando o algoritmo de eliminação de Gauss sem troca de linhas ou colunas à matriz de síndromes bi-dimensionais S, as discrepâncias serão os pivôs da matriz resultante. Portanto, o número total de discrepâncias, denotado por DT, é igual ao posto de S. Além disso, a matriz S pode ser escrita na forma $S = XYX^T$, em que $X = \|g_i(P_{u_j})\|_{k\times t}$ e $Y = \mathrm{diag}(e_{u_1}, \ldots, e_{u_t})$ (matriz diagonal). Portanto, o número total de discrepâncias é, no máximo, igual ao número de erros ocorridos t.

3.3.4 TOMADA DE DECISÃO

Considere $\mathbf{v} = \mathbf{c} + \mathbf{e}$ uma palavra recebida com $t \leq \frac{n_r-1}{2}$ erros com relação ao código C_{m_k} (n_r é definido abaixo pela equação 21). Então, todas as síndromes bi-dimensionais $S_{i,j}$, com $m_i + m_j \leq m_k$, são conhecidas e as demais síndromes são desconhecidas.

Denote o número de discrepâncias conhecidas por K. Um candidato é incorreto se e só se for uma discrepância. Então,

$$K + F \leq DT \leq t. \tag{20}$$

Se (i,j) é uma discrepância conhecida, então todas as entradas (i,j') e (i',j), com $j' > j$ e $i' < i$, não são candidatos. Se $(i,j) \in N_k$ não é um candidato, então existe pelo menos uma discrepância conhecida na mesma linha i ou coluna j. Portanto, o número de pares $(i,j) \in N_k$ que não são candidatos é, no máximo, $2K$.

O número de pares $(i,j) \in N_k$ que são candidatos é igual a $T + F$. Portanto,

$$\begin{aligned} n_r &= \text{n}^\text{o} \text{ de candidatos} + \text{n}^\text{o} \text{ de não candidatos} \\ &\leq (T+F) + 2K. \end{aligned} \tag{21}$$

Como, por hipótese, $w(\mathbf{e}) = t \leq \frac{n_r-1}{2}$, tem-se das inequações 20 e 21 que $K + F \leq \frac{n_r-1}{2} \Rightarrow 2K + 2F + 1 \leq n_r \leq T + F + 2K \Rightarrow F < T$.

Não há uma forma direta de determinar se um candidato é ou não verdadeiro. Contudo, se for atribuído um valor predito a cada um dos candidatos, então a maioria, T candidatos, terá o mesmo valor, que é por definição o valor correto de S_{r+1}.

Um algoritmo de decodificação com decisão por maioria na forma apresentada aqui tem uma complexidade $\mathcal{O}(n^3)$, no entanto algoritmos que associam o algoritmo BMS ao esquema de decisão por maioria apresentam complexidades menores, da ordem de $\mathcal{O}\left(n^{\frac{7}{3}}\right)$.

3.4 ALGORITMO DE PORTER (SEGUNDA ABORDAGEM)

O algoritmo de decodificação de Porter pode ser visto como uma generalização da solução da equação chave para códigos de Goppa clássicos pelo algoritmo de Euclides em um anel de polinômios em uma única variável [27]. No caso dos códigos AG, ao invés de um anel de polinômios em uma variável, tem-se o anel de funções racionais em uma curva, denotado por $K_\infty(P)$, definido a seguir.

3.4.1 ANEL AFIM $K_\infty(P)$

Considere uma curva projetiva, não singular e irredutível $\mathcal{X}$ de gênero g definida sobre o corpo $\mathbb{F}_q$. Considere $\mathbb{F}_q(\mathcal{X})$ o corpo das funções racionais em $\mathcal{X}$. Considere $P, P_1, \ldots, P_n$ pontos racionais (equivalentes a lugares de grau 1) de $\mathcal{X}$[5], e D o divisor $P_1 + \cdots + P_n$. Considere também o divisor G, cujo suporte é disjunto de $\{P_1, \ldots, P_n\}$.

Defina, então, o anel afim $K_\infty(P)$ com relação ao ponto (lugar) P como sendo $K_\infty(P) = \{f \in \mathbb{F}_q(\mathcal{X}) \mid \sup((f)_\infty) \subseteq \{P\}\}$, ou seja, o conjunto das funções racionais em $\mathcal{X}$ que possuem pólos exclusivamente em P.

Defina o *grau de uma função* $f \in K_\infty(P)$ por $\deg(f) = -v_P(f)$, em que $v_P(f)$ é a função de avaliação discreta de f em P. Observe que, se $f, g \in K_\infty(P)$, então $\deg(fg) = \deg(f) + \deg(g)$, e $\deg(f+g) \leq \max[\deg(f), \deg(g)]$. Também, se $\deg(f) = \deg(g)$, então existe um $\lambda \in \mathbb{F}_q^*$, tal que $\deg(f - \lambda g) < \deg(f)$.

3.4.2 ISOMETRIA DE CÓDIGOS

Considere um código linear C. Se $\mathbf{c} = (x_1, \ldots, x_n)$ é uma palavra código de C, defina $\sigma\mathbf{c} = (x_{\sigma(1)}, \ldots, x_{\sigma(n)})$ como sendo uma permutação das posições da palavra $\mathbf{c}$, e $\sigma C = \{\sigma\mathbf{c} \mid \mathbf{c} \in C\}$. Dois códigos lineares C_1 e C_2 em $\mathbb{F}_q^n$ são ditos equivalentes se $C_1 = \sigma C_2$, para alguma permutação σ. Considere $\lambda = (\lambda_1, \ldots, \lambda_n) \in \mathbb{F}_q^n$ uma n-úpla não nula. Defina, então, $\lambda\mathbf{c} = (\lambda_1 x_1, \ldots, \lambda_n x_n)$ e $\lambda C = \{\lambda\mathbf{c} \mid \mathbf{c} \in C\}$. Dois códigos lineares C_1 e C_2 em $\mathbb{F}_q^n$ são ditos *isométricos* se existir uma n-úpla λ de elementos não nulos de $\mathbb{F}_q$ e uma permutação σ, tais que $C_1 = \lambda\sigma C_2$. Pode-se ver que este mapeamento $\lambda\sigma$ mantém a métrica de Hamming do código invariante.

Considere C_1 e C_2 dois códigos isométricos em $\mathbb{F}_q^n$, com $C_1 = \lambda\sigma C_2$. Suponha que $A(C_2)$ é um algoritmo de decodificação de C_2 que corrige até t erros. O seguinte procedimento, chamado *algoritmo de decodificação induzido* $\lambda\sigma A(C_2)$, corrige o código C_1 também até t erros:

1. Entrada: $\mathbf{v}$;

[5]Neste algoritmo, é necessário reservar um dos pontos racionais da curva para a definição de um divisor extra E.

2. $\mathbf{u} = \sigma^{-1}\left(\frac{x_1}{\lambda_1}, \ldots, \frac{x_\mu}{\lambda_n}\right)$;

3. Execute $A(C_2)$ com o vetor de entrada u para obter $\mathbf{c}' \in C_2$;

4. Saída: $\mathbf{c} = \lambda\sigma\mathbf{c}'$.

Portanto, uma vez que um decodificador para um dos códigos de uma classe de isometria é dado, então todos os decodificadores dos códigos desta classe são obtidos através de algoritmos induzidos.

Se m é um inteiro, então existe uma função $h \in K_\infty(P)$ e um inteiro positivo μ, tais que os códigos AG $C^*(D, mP)$ e $C^*(D, (h)_0 - \mu P)$ são isométricos. Sem perda de generalidade, portanto, será utilizado aqui o código $C^*(D, G)$, em que $G = E - \mu P$ e E é um divisor efetivo.

3.4.3 RESÍDUOS DE DIFERENCIAIS

Considere P um ponto racional de $\mathcal{X}$ disjunto de $\{P_1, \ldots, P_n\}$. Considere E um divisor efetivo e μ um inteiro positivo, tais que E e $D = P_1 + \cdots + P_n$ possuem suportes disjuntos e $\deg(E - \mu P) \geq 2g - 1$ (teorema de Riemman-Roch).

Mostra-se que existem sempre n diferenciais $\varepsilon_1, \ldots, \varepsilon_n \in \Omega(-D - \mu P)$ (espaço de diferenciais gerado pelo divisor $-D - \mu P$) independentes módulo $\Omega(-\mu P)$, tais que $\mathrm{Res}_{P_i}(\varepsilon_j) = 1$, se $i = j$, e $\mathrm{Res}_{P_i}(\varepsilon_j) = 0$, se $i \neq j$ [27]. Se, além disso, $\mu = 1$, então $(\varepsilon_i)_\infty = P_i + P$, para $1 \leq i \leq n$.

Mostra-se também que, para todo diferencial $\omega \in \Omega(E - \mu P - D)$, $\omega = \sum_{j=1}^{n} \mathrm{Res}_{P_j}(\omega)\varepsilon_j$ [27].

Defina $\varepsilon(\mathbf{c}) = \sum_i c_i\varepsilon_i$. Este mapeamento $\varepsilon : \mathbb{F}_q^n \to \Omega_\mathcal{X}$ ($\Omega_\mathcal{X}$ é o espaço de diferenciais associados à curva $\mathcal{X}$) é na verdade um mapeamento inverso de Res_D, uma vez que $\mathrm{Res}_D(\varepsilon(\mathbf{c})) = \mathbf{c}$. Além disso, $\varepsilon(\mathbf{c}) \in \Omega(E - \mu P - D)$ se e só se $\mathbf{c} \in C^*(D, E - \mu P)$.

3.4.4 AS SÍNDROMES

Segundo a primeira abordagem na decodificação de códigos AG, as síndromes de uma palavra recebida $\mathbf{v} \in \mathbb{F}_q^n$ são definidas por um mapeamento S do espaço de funções $L(G)$ para o corpo $\mathbb{F}_q$. Este mapeamento é dado por $S(\mathbf{v}, f) = \sum_{i=1}^{n} v_i f(P_i)$, em que $f \in L(G)$.

Nesta segunda abordagem, a síndrome de uma palavra recebida $\mathbf{v} \in \mathbb{F}_q^n$ é definida como um elemento do anel afim $K_\infty(P)$, que consiste numa generalização das síndromes dos códigos de Goppa clássicos. A definição das síndromes que será apresentada é valida para códigos da forma $\mathbf{C}^*(D, E - \mu P)$, o que não constitui uma restrição, uma vez que qualquer código AG é isométrico a algum código deste tipo.

Suponha que $E = (h)_0$ (divisor dos zeros de h), com $h \in K_\infty(P)$, em que h não possui zeros em qualquer dos pontos $P_1, \ldots, P_n$ de $\mathcal{X}$.

Mostra-se que existe sempre um diferencial η, tal que [27]

$$\sup((\eta)_0) \subseteq \{P\} \Rightarrow (\eta)_0 = lP \qquad (22)$$

e $\sup((\eta)) \cap (\{P_1, \ldots, P_n\} \cup \sup(E)) = \varnothing$. Se $\mathcal{X}$ é uma curva de gênero $g > 1$, então $l > 0$.

Definição 10 (Síndrome) *A síndrome de um vetor recebido* $\mathbf{v} \in \mathbb{F}_q^n$ *para um código* $\mathbf{C}^*(D, E - \mu P)$ *é definida como o mapeamento linear* $S : \mathbb{F}_q^n \to \mathbb{F}_q(\mathcal{X})$ *dado por* $S(\mathbf{v})\eta = \sum_{i=1}^{n} v_i \frac{h(P_i) - h}{h(P_i)}\varepsilon_i$.

O nome síndrome para este mapeamento S é justificado pelo fato de que, se $E = (h)_0$, então $\mathbf{v} \in C^*(D, E - \mu P) \Leftrightarrow S(\mathbf{v}) \equiv 0 \bmod h$.

Como já foi afirmado, esta síndrome $S(\mathbf{v})$ constitui um elemento de $K_\infty(P)$.

3.4.5 DECODIFICAÇÃO PELA SOLUÇÃO DA EQUAÇÃO CHAVE

Por simplicidade, assuma que o diferencial η é tal que $(\eta) = (2g - 2)P$.

Considere W um divisor em $\mathcal{X}$, tal que o ponto P não pertence ao suporte de W. Defina o ideal $K_\infty(P, W)$ como sendo $K_\infty(P, W) = \{f \in K_\infty(P) \mid f = 0 \text{ ou } v_Q(f) \geq n_Q(W)\}$, em que $n_Q(W)$ é o coeficiente do ponto Q no divisor W.

Considere $E = (h)_0$ (suporte disjunto de $\{P_1, \ldots, P_n\}$). Dada a síndrome $S(\mathbf{v})$ da palavra recebida $\mathbf{v} \in \mathbb{F}_q^n$, a decodificação de $\mathbf{v}$ é feita pela solução da *equação chave*

$$\begin{aligned} &fS(\mathbf{v}) \equiv r \bmod h \Rightarrow \\ &fS(\mathbf{v}) = r + qh, \end{aligned} \qquad (23)$$

em que $f \in K_\infty(P)$, $r, q \in K_\infty\left(P, (\eta)_{\in fty}\right)$ e $\deg(r) \leq \deg(f) + 2g - 2 + \mu$. O par (f, r) é dito ser uma *solução válida* para a equação chave. Uma solução válida (f, r) é dita ser *mínima* se $\deg(f)$ for o menor entre os graus de todas as funções f', tais que (f', r') também é uma solução válida.

Defina agora o *defeito de Clifford* σ do par (E, P) pela equação $\sigma = \max\{\frac{\deg(E - kP)}{2} - (l(E - kP) - 1) \mid k \in \mathbb{N}\}$. Mostra-se que $\sigma \leq \frac{g}{2}$ [24].

Segue, então, o chamado teorema da decodificação.

Teorema 11 (Decodificação) *Considere a palavra recebida* $\mathbf{v} = \mathbf{c} + \mathbf{e}$, *em que* $\mathbf{c} \in C^*(D, E - \mu P)$ *é uma palavra código e* $\mathbf{e} \in \mathbb{F}_q^n$ *é um vetor de erros ocorridos. Tem-se que:*

1. *(Existência) Existe uma solução válida* (f, r) *para a equação chave de* $\mathbf{v}$ *(equação 23), tal que* $\frac{r}{f}\eta \in \Omega(-D - \mu P)$ *e* $\mathbf{e} = \mathrm{Res}_D\left(\frac{r}{f}\eta\right)$;

2. *(Unicidade) Considere a ocorrência de até* $t = \frac{d-1}{2} - \sigma$ *erros, em que* d *é a distância mínima projetada do código e* σ *o defeito de Clifford. Se* (f, r) *é uma solução válida mínima da equação chave de* $\mathbf{v}$, *então* $\frac{r}{f}\eta \in \Omega(-D - \mu P)$ *e* $\mathbf{e} = \mathrm{Res}_D\left(\frac{r}{f}\eta\right)$.

Este teorema estabelece que o problema da decodificação resume-se à obtenção de uma solução válida para a equação chave (equação 23). Esta solução para a equação chave pode ser obtida através do algoritmo proposto por Shen e Tzeng [28], que utiliza uma *seqüência de sub-resultantes* e constitui uma generalização do algoritmo de Euclides. A utilização deste algoritmo de Shen associado ao algoritmo de Porter permite a decodificação dos $\frac{d-1}{2} - \sigma$ erros com uma complexidade $\mathcal{O}(n^3)$.

3.5 ALGORITMO DE SHOKROLLAHI E WASSERMAN (TERCEIRA ABORDAGEM)

Usualmente, se o número de erros inseridos no vetor recebido durante uma transmissão for maior que $\frac{d-1}{2}$, então a decodificação da palavra código única geralmente não é possível. Entretanto, para uma quantidade limitada e de erros, é possível se construir um algoritmo que forneça uma lista com todas as palavras código cujas distâncias de Hamming do vetor recebido sejam menores ou iguais a e. A este tipo de decodificação dá-se o nome de *decodificação de lista.*

Definição 12 *Um código de bloco linear C de comprimento n definido sobre $\mathbb{F}_q$ é dito (e, b)-decodificável se toda esfera de Hamming de raio e em $\mathbb{F}_q^n$ contiver no máximo b palavras código.*

Portanto, um código (e, b)-decodificável permite uma decodificação de lista com listas de tamanho no máximo b. Um mesmo código pode ser (e, b)-decodificável para diferentes valores de e e b. Para dar dois exemplos extremos, observe que qualquer código definido sobre $\mathbb{F}_q$ de comprimento n, dimensão k e distância mínima d (um código (n, k, d)) é $(\lfloor \frac{d-1}{2} \rfloor, 1)$-decodificável, assim como também (n, q^k)-decodificável.

Considere $\mathcal{X}$ uma curva algébrica de gênero g definida sobre $\mathbb{F}_q$, e $\mathbb{F}_q(\mathcal{X})$ seu corpo de funções.

Considere $P_1, \ldots, P_n$ pontos racionais da curva $\mathcal{X}$, e G um divisor de $\mathcal{X}$ cujo suporte seja disjunto do suporte do divisor $D = P_1 + \cdots + P_n$. Denote por α o grau do divisor G e considere $\alpha < n$.

Considere o espaço de funções $L(G)$ do divisor G. O chamado teorema de Riemann estabelece que a dimensão $l(G)$ de um espaço $L(G)$ sobre $\mathbb{F}_q$ é finita e limitada inferiormente por $\alpha - g + 1$.

Os códigos AG utilizados no algoritmo descrito aqui são os construídos por funções (códigos $C(D, G)$). Observe que comumente os códigos AG utilizados nos diversos algoritmos de decodificação propostos na literatura são aqueles construídos por diferenciais, diferentemente do que ocorre neste caso. Esta diferença, entretanto, não é essencial, uma vez que qualquer código AG pode ser definido das duas maneiras.

Pela aplicação do teorema de Riemann, prova-se que $C(D, G)$ é um código (n, k, d), em que $k \geq \alpha - g + 1$ e $d \geq n - \alpha$. O valor $d^* = n - \alpha$ é a distância mínima projetada de $C(D, G)$.

O algoritmo 14 de decodificação de lista é resultado do teorema que segue.

Teorema 13 *Considere C um código AG de comprimento n e dimensão k definido por uma curva algébrica $\mathcal{X}$ de gênero g sobre $\mathbb{F}_q$. Então, para um inteiro positivo b, C é um código $(n-\beta-1, b)$-decodificável, sendo $\beta = \left\lceil \frac{n+1}{b+1} + \frac{b\alpha}{2} + g - 1 \right\rceil$ e $\alpha = k + g - 1$.*

Prova. Veja [42]. ■

Algoritmo 14 (Decodificador de lista)

Entradas:

- *Um vetor recebido* $\mathbf{y} = (y_1, y_2, \ldots, y_n)$;
- *Um divisor F de grau* $\beta - b\alpha = \left\lceil \frac{n+1}{b+1} + \frac{b\alpha}{2} + g - 1 \right\rceil$, *cujo suporte seja disjunto de D.*

Saídas:

- *Uma lista de b palavras código* $\mathbf{x}$ *com distância de Hamming no máximo $n - \beta - 1$ do vetor recebido* $\mathbf{y}$.

<<< **Passo 1 – Interpolação** >>>

Encontrar um polinômio diferente de zero $H(T) = u_b T^b + \cdots + u_1 T + u_0 \in \mathbb{F}_q(\mathcal{X})[T]$, em que $u_i \in L(F + (b-j)G)$, tal que $H(P_i, y_i) = \sum_{j=0}^{b} u_j(P_i) y_i^j$ é zero para $i = 1, \ldots, n$.

<<< **Passo 2 – Fatoração** >>>

Encontrar todas as raízes ρ de $H(T)$ em $\mathbb{F}_q(\mathcal{X})[T]$. Para cada ρ encontrada, calcular o vetor $x_\rho = (\rho(P_1), \ldots, \rho(P_n))$. Se x_ρ é não definido ou se a distância entre x_ρ e o vetor recebido $\mathbf{y}$ *for maior que $n - \beta - 1$, descartar x_ρ. Caso contrário, acrescentar x_ρ à lista de palavras código de saída.*

Prova. Para detalhes sobre a prova deste algoritmo, veja [42]. ■

Teorema 15 *Considere C um código AG de comprimento n e dimensão k definido por uma curva algébrica $\mathcal{X}$ de gênero g. Considere $\alpha = k + g - 1$ e $\beta = \lceil \sqrt{2\alpha n} + g - 1 \rceil$. Então, C é $\left(n - \beta - 1, \left\lceil \sqrt{2n/\alpha} \right\rceil\right)$-decodificável.*

Prova. Veja [42]. ■

O passo 1 do algoritmo 14 consiste basicamente da solução de um sistema de equações lineares. A existência de um polinômio não trivial $H(T)$ no algoritmo segue do fato de que um sistema de equações homogêneo com mais variáveis desconhecidas que equações possui sempre uma solução não trivial.

A complexidade do algoritmo 14 do decodificador de lista é determinada pelo passo 2. A principal tarefa do algoritmo 14 consiste na determinação das raízes em $\mathbb{F}_q(\mathcal{X})$ de $H(T) \in \mathbb{F}_q(\mathcal{X})[T]$. Para completar o decodificador de lista, portanto, é necessário descrever um algoritmo que determine estas raízes. Shokrollahi e Wasserman propõem um algoritmo para fatorar completamente $H(T)$ [42]. Este algoritmo de fatoração, no entanto, não será descrito aqui.

3.6 ALGORITMO GMD (QUARTA ABORDAGEM)

Um sistema de comunicação digital possui no centro de sua topologia um canal analógico. Cabe ao modulador (demodulador) converter este canal analógico num canal digital a ser visto e acessado pelo codificador (decodificador). Um sistema de codificação para controle de erros pode apresentar melhores resultados se houver algum tipo de iteração entre o codificador (decodificador) e o modulador (demodulador). Os algoritmos de decodificação usando decisão suave, no caso o algoritmo GMD ("generalized minimum-distance"), são esquemas deste tipo, que utilizam informações extras obtidas do demodulador. Para um determinado código, a decodificação com decisão suave apresenta um desempenho melhor que a decodificação com decisão brusca ("hard decision"). No entanto, estes sistemas são mais complexos, sendo

úteis em geral apenas para códigos pequenos. Num caso extremo em que a complexidade do sistema possa ser desconsiderada, as tarefas de demodulação e decodificação podem ser feitas simultaneamente.

O algoritmo GMD, proposto por G. D. Forney Jr. em 1966, consiste num esquema genérico que pode ser aplicado a qualquer código de bloco linear. Uma discussão detalhada do algoritmo GMD, que é a base para este tipo de abordagem, pode ser obtida em Blahut [23, pp. 464–473]. Aqui, apenas como referência, segue uma descrição do algoritmo GMD.

Algoritmo 16 (Algoritmo GMD)

Entradas:

- *Um vetor recebido* $\tilde{\mathbf{v}}$ *de comprimento* n*;*
- *Um nível de confiabilidade* α *associado a cada posição do vetor* $\tilde{\mathbf{v}}$*.*

Saídas:

- *Uma palavra código decodificada* $\tilde{\mathbf{c}}_r$*, ou uma indicação de falha de decodificação.*

<<< **Passo 1** >>>

Determinar as $d^* - 1$ *componentes* $\tilde{v}_i$ *do vetor recebido* $\tilde{\mathbf{v}}$ *que apresentam os menores níveis de confiabilidade* α*, ordenadas segundo estes:* $\alpha_{i_1} \leq \alpha_{i_2} \leq \alpha_{i_3} \leq \cdots \leq \alpha_{i_{d^*-1}}$*. Fazer* $l = 0$*.*

<<< **Passo 2** >>>

Introduzir apagamento nas l *componentes do vetor recebido* $\tilde{\mathbf{v}}$ *de menores níveis de confiabilidade* α*, ou seja,* $\tilde{v}_{i_1}, \tilde{v}_{i_2}, \ldots, \tilde{v}_{i_l}$*. Executar o decodificador de erros e apagamentos para o vetor recebido* $\mathbf{v}$ *acrescido dos apagamentos.*

<<< **Passo 3** >>>

Verificar se $\tilde{\mathbf{v}} \cdot \tilde{\mathbf{c}}_r > n - d^*$*. Se a desigualdade for verdadeira, concluir que a palavra* $\tilde{\mathbf{c}}_r$ *fornecida pelo decodificador de erros e apagamentos é a palavra código procurada e parar o algoritmo. Se a desigualdade for falsa ou se o decodificador de erros e apagamentos não fornecer uma palavra* $\tilde{\mathbf{c}}_r$ *decodificada, fazer* $l = l + 2$*.*

<<< **Passo 4** >>>

Verificar se $l > d^* - 1$*. Se a desigualdade for verdadeira, concluir que ocorreu falha na decodificação e parar o algoritmo. Se a desigualdade for falsa, retornar ao passo 2.*

Prova. Para maiores detalhes sobre a prova deste algoritmo, veja [23, pp. 464–473]. ■

Baseado nas informações de decisão suave, o algoritmo GMD (algoritmo 16) gera uma série de vetores $\tilde{\mathbf{v}}^{(l)}$ incluindo apagamentos ao vetor recebido $\tilde{\mathbf{v}}$. Para cada l, de 0 a $d^* - 1$, ele apaga as l componentes da palavra recebida para as quais o nível de confiabilidade é menor. O vetor $\tilde{\mathbf{v}}^{(l)}$ é, então, decodificado usando um decodificador de erros e apagamentos. Se o decodificador fornecer uma palavra código $\tilde{\mathbf{c}}_r$, realiza-se o teste $\tilde{\mathbf{v}} \cdot \tilde{\mathbf{c}}_r > n - d^*$. Se este teste for satisfeito, então $\tilde{\mathbf{c}}_r$ é a palavra código decodificada. Caso contrário, o valor de l é incrementado em dois e o laço do algoritmo é repetido enquanto $l \leq d^* - 1$. Caso nenhuma palavra código seja encontrada, um falha de decodificação é declarada.

Observe que a complexidade do algoritmo GMD é aproximadamente igual a $\frac{1}{2}(d^* - 1)$ vezes a complexidade do decodificador para erros e apagamentos usado para o mesmo código.

A decodificação de códigos AG usando decisão suave é feita diretamente utilizando o algoritmo GMD, acrescentando-se um decodificador de erros e apagamentos para códigos AG no passo 2 do algoritmo 16. Diversas publicações têm apresentado derivações mais eficientes do decodificador GMD para códigos RS e códigos AG [36, 48–50].

4. IMPLEMENTAÇÃO EM HARDWARE

O desenvolvimento de algoritmos de decodificação associados a arquiteturas de implementação em hardware é de fundamental importância para tornar os códigos AG competitivos quando comparados, por exemplo, com códigos BCH não binários ou códigos concatenados. Desta forma, na implementação em hardware, não apenas a complexidade computacional do algoritmo deve ser considerada, mas também a parte de controle necessária à operacionalização do algoritmo, a interligação de elementos no circuito, além dos requisitos de espaço.

São poucos os artigos publicados tratando da questão da implementação em hardware de decodificadores para códigos AG. Em 1997, M. E. O'Sullivan e S. P. Pope apresentaram resultados sobre algoritmos e arquiteturas de implementação de códigos AG que demonstram a viabilidade dos decodificadores para estes códigos [52].

O principal trabalho relativo a este tema foi publicado em 1998 por R. Kötter, que consiste numa versão do algoritmo BMS voltada à implementação em hardware [53]. A idéia central nesta proposta de Kötter é utilizar uma configuração em paralelo de vários algoritmos modificados de Berlekamp-Massey, de modo a obter um esquema que apresente a mesma complexidade de tempo que um decodificador de Berlekamp-Massey para códigos RS.

Em 2001, J. B. Ashbrook et al propuseram uma implementação em circuito integrado CMOS de um decodificador de códigos AG definidos sobre curvas de Hermite a partir do trabalho de Kötter [54]. Esta foi a primeira implementação em hardware de um decodificador para códigos AG. Este decodificador de códigos AG sobre curvas de Hermite proposto trabalha com códigos de comprimento $n = 4080$, tem capacidade de correção de 60 erros em $\mathbb{F}_{256}$ e proporciona um ganho de codificação de 0,6 dB com relação a um código RS de mesma taxa. Ainda em 2001, E. M. Popovici et al propuseram também uma implementação de um decodificador de códigos AG sobre curvas de Hermite baseado no algoritmo de Kötter [55].

Outro trabalho importante nesta área foi apresentado em 1999 por C.-W. Liu et al [56], que propuseram uma implementação em hardware do algoritmo proposto por Feng e Rao em [31] baseada numa estrutura de arranjo sistólico. Entretanto, em [57] Z. M. Belkoura apontou incorreções nas equações que determinam a estrutura sistólica proposta, comprometendo as vantagens obtidas por esta estrutura. Em [58], apresenta-se uma arquitetura considerando as equações corrigidas.

Além dos esforços para obtenção de arquiteturas para implementação eficiente de decodificadores para códigos AG, é importante destacar também a necessidade de esquemas de implementação eficiente das diversas operações em corpo finito, uma vez que a aritmética de corpo finito é o pano de fundo para todos os algoritmos de codificação / decodificação. As operações aritméticas num corpo finito $\mathbb{F}_{2^m}$ são um pouco distintas das operações com números binários. Os elementos de $\mathbb{F}_{2^m}$ podem ser representados a partir de uma base. Usando esta representação de base, as operações de adição e subtração tornam-se simples, contrariamente às operações de multiplicação e divisão. Diversos algoritmos existem para uma implementação mais eficiente destas operações [1, 59–64].

O principal aspecto a ser observado no que tange o desenvolvimento de decodificadores voltados à implementação em hardware é a necessidade de estruturas que permitam a realização de operações em paralelo, o que proporciona uma redução na complexidade temporal do algoritmo (em detrimento da complexidade espacial). Outro aspecto importante é o desenvolvimento de estruturas simples e regulares. A regularidade da estrutura proporciona uma menor complexidade espacial e uma redução da complexidade de controle e interligação de elementos na implementação.

5. CONCLUSÕES

Este texto teve por objetivos contextualizar o problema da decodificação dos códigos AG e estabelecer um panorama das diferentes abordagens de decodificação, além da questão do desenvolvimento de arquiteturas de implementação em hardware para estes decodificadores.

Das abordagens descritas, a mais explorada tem sido a primeira, baseada no algoritmo básico. Merece destaque o trabalho de R. Kötter [53], que resultou numa primeira implementação em hardware de um decodificador para códigos AG definidos sobre curvas de Hermite [54], constituindo um primeiro passo para a viabilidade comercial destes códigos.

É importante observar que, das abordagens estudadas, apresentam grande relevância para o estudo e desenvolvimento de decodificadores voltados à uma implementação eficiente em hardware os algoritmos que se utilizam do conceito das bases de Gröbner [29, 30, 37, 65–67], uma vez que esta teoria sugere estruturas mais regulares e apresenta um forte apelo computacional. O algoritmo BMS é um exemplo deste tipo de esquema, já que o conjunto dos polinômios localizadores de erros obtido pelo algoritmo é na verdade uma base de Gröbner mínima para o ideal de funções que obedecem à relação de recursão linear definida pelas síndromes do vetor recebido.

AGRADECIMENTOS

Este trabalho foi financiado pela Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – CAPES (http://www.capes.gov.br).

REFERÊNCIAS

[1] Stephen B. Wicker and Vijay K. Bhargava. *Reed-Solomon codes and their applications*. Piscataway, NJ, USA: IEEE Press, 1994.

[2] P. J. W. A. M. Hamouda. Space-time mmse multiuser detection in multipath channels with rs coding. In *IEEE International Conference on Communications*, 2001.

[3] V. D. Goppa. A new class of linear error-correcting codes. *Problems of Information Theory*, 6:207–212, 1970.

[4] M. A. Tsfasman, S. G. Vlăduţ, and T. Zink. Modular curves, Shimura curves and Goppa codes, better than Varshamov-Gilbert bound. *Math. Nachr.*, 104:13–28, 1982.

[5] Jacobus H. van Lint and T. A. Springer. Generalized Reed-Solomon codes from algebraic geometry. *IEEE Transactions on Information Theory*, 33(3):305–309, May 1987.

[6] C. X. S. Ling. A class of linear codes with good parameters from algebraic curves. *IEEE Transactions on Information Theory*, 46:1527–1532, 2000.

[7] R. E. Blahut. Encoding of codes on curves. In *IEEE International Symposium on Information Theory*, 1994.

[8] D. Cunsheng, H. Niederreiter, and X. Chaoping. Some new codes from algebraic curves. *IEEE Transactions on Information Theory*, 46:2638–2642, 2000.

[9] M. Elia, E. Viterbo, and G. Bertinetti. Decoding of binary separable Goppa codes using Berlekamp-Massey algorithm. *Electronics Letters*, 35:1720–1721, 1999.

[10] Francisco Marcos de Assis. Hit probability between frequency hopping sequences generated by Reed-Solomon and Hermitian codes. *Electronics Letters*, 32(11):962–963, May 1996.

[11] Chaoping Xing. Algebraic-geometry codes with asymptotic parameters better than the Gilbert-Varshamov and the Tsfasman-Vlăduţ-Zink bounds. *IEEE Transactions on Information Theory*, 47:347–352, January 2001.

[12] William Fulton. *Algebraic Curves: An Introduction to Algebraic Geometry*. Reading, MA: W. A. Benjamin, 1969.

[13] David Cox, John Little, and Donal O'Shea. *Ideals, Varieties, and Algorithms: An Introduction to Computational Algebraic Geometry and Commutative Algebra*. New York: Springer-Verlag, 1992.

[14] David A. Cox, John B. Little, and Donal B. O'Shea. *Using Algebraic Geometry*. New York: Springer-Verlag, 1998.

[15] Judy L. Walker. *Codes and Curves*. Oxford University Press, 2000.

[16] Oliver Pretzel. *Codes and Algebraic Curves*. New York: Oxford University Press, 1998.

[17] Henning Stichtenoth. *Algebraic Function Fields and Codes*. Berlin: Springer-Verlag, 1993.

[18] Alfred J. Menezes, editor. *Applications of Finite Fields*. Boston: Kluwer Academic Publishers, 1993.

[19] Leocarlos B. S. Lima. Análise dos algoritmos de decodificação para códigos de geometria algébrica sobre curvas de Hermite. Master's thesis, Universidade Federal da Paraíba – UFPB, August 1999.

[20] Ian Blake, Chris Heegard, Tom Høholdt, and Victor Wei. Algebraic-geometry codes. *IEEE Transactions on Information Theory*, 44(6):2596–2618, October 1998.

[21] Jørn Justesen, Knud J. Larsen, H. Elbrønd Jensen, Allan Havemose, and Tom Høholdt. Construction and decoding of a class of algebraic geometric codes. *IEEE Transactions on Information Theory*, 35(4):811–821, July 1989.

[22] Jacobus H. Van Lint. Algebraic geometric codes. In *Coding Theory and Design Theory (Part I)*, volume 21 of *IMA Volumes Math. Appl.*, pages 137–162. Berlin: Springer-Verlag, 1990.

[23] Richard E. Blahut. *Theory and Practice of Error Control*

Codes. Reading, MA: Addison-Wesley, 1983.

[24] Alexei N. Skorobogatov and Sergei G. Vlăduţ. On the decoding of algebraic-geometric codes. *IEEE Transactions on Information Theory*, 36(5):1051–1060, September 1990.

[25] Leocarlos B. S. Lima and Francisco M. Assis. Decoding algorithm for Reed-Solomon codes using the method of Gröbner basis. In *Proc. 2nd Conference on Telecommunications – ConfTele99*, pages 350–353, Sesimbra, Portugal, 1999.

[26] Patrick Fitzpatrick. On the key equation. *IEEE Transactions on Information Theory*, 41(5):1290–1302, September 1995.

[27] S. C. Porter, Ba-Zhong Shen, and Ruud Pellikaan. Decoding geometric Goppa codes using an extra place. *IEEE Transactions on Information Theory*, 38(6):1663–1676, November 1992.

[28] Ba-Zhong Shen and Kenneth K. Tzeng. Decoding geometric Goppa codes up to designed minimum distance by solving a key equation in a ring. *IEEE Transactions on Information Theory*, 41(6):1694–1702, November 1994.

[29] Michael E. O'Sullivan. Decoding of codes defined by a single point on a curve. *IEEE Transactions on Information Theory*, 41(6):1709–1719, November 1995.

[30] Michael E. O'Sullivan. Decoding of Hermitian codes: The key equation and efficient error evaluation. *IEEE Transactions on Information Theory*, 46(2):512–523, March 2000.

[31] Gui-Liang Feng and T. R. N. Rao. Decoding of algebraic geometric codes up to the designed minimum distance. *IEEE Transactions on Information Theory*, 39(1):37–45, January 1993.

[32] Shajiro Sakata. Extension of the Berlekamp-Massey algorithm to N dimensions. *Informat. Comput.*, 84:207–239, February 1990.

[33] Shojiro Sakata, Yukio Numakani, and Masaya Fujisawa. A fast interpolation method for list decoding of RS and algebraic-geometric codes. In *ISIT Proceedings*, page 479, 2000.

[34] C. D. Jensen. Fast decoding of codes from algebraic geometry. *IEEE Transactions on Information Theory*, 40:223–230, January 1994.

[35] Jørn Justesen, Knud J. Larsen, H. Elbrønd Jensen, and Tom Høholdt. Fast decoding of codes from algebraic plane curves. *IEEE Transactions on Information Theory*, 38(1):111–119, January 1992.

[36] Ralf Kötter. Fast generalized minimum-distance decoding of algebraic-geometry and Reed-Solomon codes. *IEEE Transactions on Information Theory*, 42(3):721–737, May 1996.

[37] Keith Saints and Chris Heegard. Algebraic-geometric codes and multidimensional cyclic codes: A unified theory and algorithms for decoding using Gröbner bases. *IEEE Transactions on Information Theory*, 41(6):1733–1751, November 1995.

[38] Shojiro Sakata, Helge Elbrond Jensen, and Tom Høholdt. Generalized Berlekamp-Massey decoding of algebraic geometric codes up to half the Feng-Rao bound. *IEEE Transactions on Information Theory*, 41:1762–1768, November 1995.

[39] Shajiro Sakata, Jørn Justesen, Y. Madelung, H. Elbrønd Jensen, and Tom Høholdt. Fast decoding of algebraic-geometric codes up to the designed minimum distance. *IEEE Transactions on Information Theory*, 41(5):1672–1677, September 1995.

[40] Shojiro Sakata, Douglas A. Leonard, Helge Elbrond Jensen, and Tom Høholdt. Fast erasure-and-error decoding of algebraic geometry codes up to the Feng-Rao bound. *IEEE Transactions on Information Theory*, 44:1558–1565, July 1998.

[41] Gui-Liang Feng, Victor K. Wei, T. R. N. Rao, and Kenneth K. Tzeng. Simplified understanding and efficient decoding of a class of algebraic-geometric codes. *IEEE Transactions on Information Theory*, 40(4):981–1002, July 1994.

[42] M. Amin Shokrollahi and Hal Wasserman. List decoding of algebraic-geometric codes. *IEEE Transactions on Information Theory*, 45:432–437, March 1999.

[43] Peter Elias. List decoding for noisy channels. *Technical Report 335, Research Laboratory of Electronics, MIT*, 1957.

[44] John M. Wozencraft. List decoding. *Quarterly Progress Report, Research Laboratory of Electronics, MIT*, 48:90–95, 1958.

[45] Venkatesan Guruswami and Madhu Sudan. Improved decoding of Reed-Solomon and algebraic-geometric codes. *IEEE Transactions on Information Theory*, 45:432–437, March 1999.

[46] Xin-Wen Wu and Paul H. Siegel. Efficient root-finding algorithm with application to list decoding of algebraic-geometric codes. *IEEE Transactions on Information Theory*, 47(6):2579–2587, September 2001.

[47] Tom Høholdt and Ruud Pellikaan. On the decoding of algebraic-geometric codes. *IEEE Transactions on Information Theory*, 41(6):1589–1614, November 1995.

[48] Elwyn R. Berlekamp. Bounded distance + 1 soft-decision Reed-Solomon decoding. *IEEE Transactions on Information Theory*, 42(3):704–720, 1996.

[49] Ralf Kötter and Alexander Vardy. Algebraic soft-decision decoding of Reed-Solomon codes. In *ISIT Proceedings*, page 61, 2000.

[50] Vishakan Ponnampalam and Branka Vucetic. Soft decision decoding of Reed-Solomon codes. In *ISIT Proceedings*, page 62, 2000.

[51] G. D. Forney Jr. Generalized minimum distance decoding. *IEEE Transactions on Information Theory*, IT-12:125–131, April 1966.

[52] Michael E. O'Sullivan and Stephen P. Pope. VLSI architecture for a decoder for Hermitian codes. In *ISIT Proceedings*, page 376, 1997.

[53] Ralf Kötter. A fast parallel implementation of a Berlekamp-Massey algorithm for algebraic-geometric codes. *IEEE Transactions on Information Theory*, 44:1353–1368, July 1998.

[54] Jonathan B. Ashbrook, Naresh R. Shanbhag, Ralf Kötter, and Richard E. Blahut. Implementation of a Hermitian decoder IC in $0.35\mu m$ CMOS. In *IEEE Custom Integrated Circuits Conference Proceedings*, pages 297–300, 2001.

[55] E. M. Popovici, M. E. O'Sullivan, P. Fitzpatrick, and Ralf Kötter. Implementation of a Hermitian decoder. In *ISIT Proceedings*, page 311, 2001.

[56] Chih-Wei Liu, Kuo-Tai Huang, and Chung-Chin Lu. A systolic array implementation of the Feng-Rao algorithm. *IEEE Transactions on Computers*, 48:690–706, 1999.

[57] Zouhair M. Belkoura. On hardware implementation of decoding for error correcting codes based on algebraic geometry. Final year project, Ecole Nationale Supérieure des Télécommunications, 2002.

[58] Zouhair M. Belkoura and Lirida Alves de Barros Naviner. Hardware implementation issues of a BMS decoding approach for AG based codes. In *Proceedings of the IEEE Wireless Communications and Networking Conference*, pages 448–453, 2003.

[59] Edoardo D. Mastrovito. *VLSI Architectures for Computations in Galois Fields*. PhD thesis, Linköping University, Sweden, 1991.

[60] Huapeng Wu, Anwarul Hasan, and Ian F. Blake. New low-complexity bit-parallel finite field multipliers using weakly dual bases. *IEEE Transactions on Computers*, 47(11):1223–1234, November 1998.

[61] Tong Zhang and Keshab K. Parhi. Systematic design of original and modified Mastrovito multipliers for general irreducible

polynomials. *IEEE Transactions on Computers*, 50(7):734–749, July 2001.

[62] Christof Paar. *Efficient VLSI Architectures for Bit Parallel Computation in Galois Fields*. PhD thesis, Fortschritt-Berichte VDI, Germany, 1994.

[63] A. Halbutogullari and C. K. Koc. Mastrovito multiplier for general irreducible polynomials. *IEEE Transactions on Computers*, 49(5):503–518, May 2000.

[64] Leocarlos B. S. Lima, Lirida A. B. Naviner, and Francisco M. Assis. Implantation matérielle d'unités arithmétiques en corps finis pour le codage de canal. In *JNRDM Proceedings*, pages 397–399, 2003.

[65] Douglas A. Leonard. A generalized Forney formula for algebraic-geometric codes. *IEEE Transactions on Information Theory*, 42:1263–1268, July 1996.

[66] Douglas A. Leonard. Efficient Forney functions for decoding AG codes. *IEEE Transactions on Information Theory*, 45:260–265, January 1999.

[67] Douglas A. Leonard. Using Gröbner basesin decoding AG codes. In *ITW Proceedings*, pages 31–32, 1998.

Leocarlos B. S. Lima formou-se Engenheiro Eletricista em 1997 pela então Universidade Federal da Paraíba – UFPB, Campus II, hoje Universidade Federal de Campina Grande – UFCG, em Campina Grande-PB, Brasil. Concluiu mestrado na área de Telecomunicações em 1999 pela mesma instituição. Entre 1999 e 2000, trabalhou pela Netcon Ltda. de Recife-PE, Brasil, como supervisor de obras de instalação de cabos e terminações ópticas da rede nacional de telefonia da Intelig. Atualmente, realiza doutorado em Telecomunicações pela UFCG e pela École Nationale Supérieure des Télécommunications – ENST, em Paris, França.

Francisco M. Assis Formação acadêmica: Arma de Comunicações pela Academia Militar das Agulhas Negras (AMAN/RJ), 1978, Engenharia Elétrica (Telecomunicações) pelo Instituto Militar de Engenharia (IME/RJ), 1985, Mestrado em Sistemas de Comunicações, pelo IME/RJ, 1991, Doutorado em Sistemas de Telecomunicações pela Pontifícia Universidade Católica do Rio de Janeiro (PUC/Rio), 1994. É professor adjunto do Departamento de Engenharia Elétrica da Universidade Federal de Campina Grande (DEE-UFCG) desde 1993. Suas áreas de interesse são teoria da informação, codificação e complexidade.

Lirida A. B. Naviner concluiu os cursos de Engenharia Elétrica e Mestrado em Processamento da Informação pela Universidade Federal da Paraíba – UFPB, respectivamente em 1988 e 1990. De 1994 a 1997, após conclusão do Doutorado em Eletrônica e Comunicações pela École Nationale Supérieure des Télécommunications – ENST, foi professora pesquisadora junto aos departamentos de Engenharia Elétrica e Sistemas de Computação da UFPB. Desde 1998, faz parte do quadro permanente de professores da ENST em Paris, França, onde exerce atividades de ensino, pesquisa e extensão. Membro do Centre National de Recherche Scientifique – CNRS, seus temas de interesse atuais são codificação de fonte/canal, filtragem para sistemas de comunicação multi-padrões, hw/sw codesign e arquiteturas reconfiguráveis e evolutivas.