Construção de Códigos de Bloco Lineares sobre $\mathbb{F}_q$ com d_{min} Máxima usando Programação Linear Inteira

Rodrigo Gusmão Cavalcante e Reginaldo Palazzo Jr.

***Resumo*— Este trabalho apresenta uma representação modular para códigos de bloco lineares sobre $\mathbb{F}_q$ com medida de distância mais geral que as conhecidas. Usando tal representação o modelo resultante é caracterizado como um problema de otimização inteira, cujas soluções são códigos de bloco sobre $\mathbb{F}_q$ com d_{min} máxima. Analisamos o desempenho de alguns dos principais métodos de otimização aplicados à formulação proposta, reproduzindo algumas classes de códigos já conhecidas e indicando a construção de possíveis novas classes. Este trabalho tem aplicação direta no projeto de codificação de sistemas de comunicações digitais em espaços mais gerais que o Euclidiano.**

***Palavras-Chave*— Códigos de Bloco Lineares, Representação Modular, Programação Linear Inteira, Espaços Métricos.**

***Abstract*— This paper establishes a modular representation for linear block codes over $\mathbb{F}_q$ with a more general distance measure than the previous ones. By using such a representation the resulting modelis characterized as an integer optimization problem whose solutions are block codes over $\mathbb{F}_q$ with maximum d_{min}. We analyze the performance of some of the main optimization methods applied to the proposed model, some classes of already known codesare reproduced as well as the construction of new classes of codes. As a consequence, the results obtained have applications inthe design of coded digital communication systems in spaces more general than the Euclidean one.**

***Keywords*— Linear Block Codes, Modular Representation, Linear Integer Programming, Metric Spaces.**

I. INTRODUÇÃO

Em teoria da codificação existe um interesse natural de encontrar códigos de bloco cuja distância mínima, d_{min}, seja máxima. Sabemos que tal parâmetro é um bom indicador do desempenho de sistemas de comunicações que usam decodificação de máxima verossimilhança e possuem relação sinal ruído de moderada a alta. Contudo, a busca de códigos de bloco com distâncias altas é um problema difícil, podendo resultar em tarefas computacionais de otimização intratáveis.

Além dos clássicos códigos de bloco de Hamming, Reed-Muller, BCH, Reed-Solomon, Goppa, etc, existem muitos outros códigos que podem ser construídos por diferentes técnicas, tais como: aumento do comprimento (*lengthening*), puncionamento (*puncturing*), soma direta (*direct sum*), concatenação (*concatenade code*), etc. Neste trabalho, apresentamos um método de construção de códigos de bloco lineares $\mathcal{C}$ sobre $\mathbb{F}_q$, q primo, de taxa $r = k/n$ com d_{min} máxima. Para tanto, utilizamos a representação modular para códigos de bloco, [10], com o intuito de formular um problema de otimização linear em função de d_{min} que possa ser resolvido por programação linear inteira, [6].

Departamento de Telemática, Faculdade de Engenharia Elétrica e de Computação, Universidade Estadual de Campinas, Campinas, Brasil, E-mails: rgc@dt.fee.unicamp.br, palazzo@dt.fee.unicamp.br.

A representação modular já havia sido usada em [3] e [9] na construção de códigos convolucionais provenientes também de um problema de otimização (*Knapsack Problem*). A programação linear, juntamente com a identidade de MacWilliams-Delsarte [4], foi usada com o propósito de provar a inexistência de códigos lineares binários,[1], e mais recentemente com o objetivo de otimizar a distância espectral de códigos de bloco lineares, [5].

Neste trabalho a distribuição de pesos de um código $\mathcal{C}$ será representada pelo seu polinômio enumerador $W_{\mathcal{C}}(t) = a_0 + a_1 t + \cdots + a_i t^i$, onde a_i é o número de palavras-código de peso i. Com o objetivo de identificar os melhores códigos de bloco corretores de erros, considere dois códigos de bloco $\mathcal{C}$ e $\mathcal{C}'$, q-ários com mesma taxa $r = k/n$, então $W_{\mathcal{C}}$ é melhor do que $W_{\mathcal{C}'}$ se $a'_i = a_i$ para $0 \leq i \leq j$ e $a'_{j+1} > a_{j+1}$. Assim, chamaremos de código ótimo todo código que possuir o melhor dos polinômios enumeradores. Por exemplo, caso $W_{\mathcal{C}} = 1 + t^2 + 2t^3$ e $W_{\mathcal{C}'} = 1 + 2t^2 + t^3$ então o código $\mathcal{C}$ será melhor do que $\mathcal{C}'$, pois $a'_i = a_i$ para $i = 0, 1$ e $a'_2 > a_2$. Conseqüentemente, o código ótimo terá o maior valor de d_{min} e a menor quantidade de palavras-código com tal distância.

Este trabalho está organizado da seguinte maneira. Na seção II, apresentamos a representação modular para códigos de blocos. Na seção III, formulamos o problema de otimização inteira cujas soluções são códigos com d_{min} máxima. Apresentamos uma medida de distância mais geral que as usadas atualmente e um limitante superior para a d_{min} de códigos com tal distância. Ainda nessa seção, construímos códigos de bloco usando alguns dos principais métodos de otimização aplicados à formulação proposta. Na seção IV, apresentamos algumas simplificações aplicadas ao problema de otimização com o intuito de propiciar a geração de classes de códigos e minimizar o esforço computacional.

II. REPRESENTAÇÃO MODULAR PARA CÓDIGOS DE BLOCO

A representação modular descrita em [10] é usada para representar apenas códigos de bloco binários com distância de Hamming. Nesta seção, vamos generalizar tal representação para códigos de bloco q-ários com medida de distância mais geral que a de Hamming.

Em geral, os códigos de bloco são descritos por uma matriz geradora G que possui k linhas e $m = q^k - 1$ possíveis tipos diferentes de colunas, uma vez que a coluna toda nula não é considerada. Como o rearranjo das colunas de uma matriz geradora G implica em um código equivalente, então um código pode ser descrito por um conjunto de colunas

denominado representação modular, isto é,

$$N = [n_1, n_2, \ldots, n_m], \tag{1}$$

onde n_i é um número inteiro que representa a quantidade de colunas do tipo i (representação binária do número i) em G e $\sum n_i = n$.

Usando a representação modular para códigos de bloco, podemos obter o espectro de pesos, W, das m palavras-código não nulas de um código de bloco por

$$W = N \cdot C, \tag{2}$$

onde N é obtida de (1) e C é uma matriz simétrica de ordem m. A matriz C será convenientemente denotada por *matriz de pesos*, pois ela depende da medida de distância do código.

Para obtermos essa matriz de pesos considere inicialmente que a matriz geradora de um código seja a matriz M, onde M é uma matriz $k \times m$ que possui como colunas todas as possíveis combinações de k elementos de $\mathbb{F}_q$, exceto a combinação toda nula. Considere também, sem perda de generalidade, que a j-ésima coluna de M seja do tipo j.

Em seguida, devemos definir a matriz $C^* = (M' \cdot M) \bmod q$ que tem como linhas todas as possíveis combinações lineares das linhas da matriz geradora e por isso possui todas as palavras-código como linhas, veja exemplo 1. Finalmente, devemos aplicar uma função distância $f_d : \mathbb{F}_q \to R^+$ em cada um dos elementos c^*_{ij} de C^* de maneira que $c_{ij} = f_d(c^*_{ij})$, obtendo assim a matriz de pesos C, veja exemplo 2.

Desse modo, temos de (2) que o peso w_i de cada uma das m palavras-código de um código de bloco é calculada por

$$w_i = \sum_j n_i c_{ij} = \sum_j n_i f_d(c^*_{ij}), \quad \text{para } i, j = 1, \ldots, m. \tag{3}$$

Portanto, note que essa representação modular pode descrever códigos de bloco em espaços métricos cuja distância é dada por (3). Isto possibilita a construção de códigos de bloco para sistemas de comunicações digitais em espaços mais gerais do que o euclidiano, como proposto em [7] e [8]. Os exemplos mais usuais de funções distância são a distância de Hamming,

$$c_{ij} = \begin{cases} 0, & \text{se } c^*_{ij} = 0 \\ 1, & \text{se } c^*_{ij} \neq 0 \end{cases}, \tag{4}$$

a distância de Lee,

$$c_{ij} = \begin{cases} c^*_{ij}, & \text{se } c^*_{ij} \leq q/2 \\ q - c^*_{ij}, & \text{se } c^*_{ij} > q/2 \end{cases}, \tag{5}$$

e a distância euclidiana ao quadrado,

$$c_{ij} = (c^*_{ij})^2. \tag{6}$$

Exemplo 1 Considere $q = 2$ e $k = 3$, então as matrizes M e C são dadas por

$$M^T = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix}, \quad C^* = \begin{bmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{bmatrix}.$$

Note que as colunas de M estão em ordem crescente e representam os números inteiros de 1 a 7. Para o caso binário, podemos construir as seguintes fórmulas recursivas, em função de k, para as matrizes M e C:

$$M_k = \left[\begin{array}{c|c|c} 0 & 1 & 1 \\ \hline M_{k-1} & 0 & M_{k-1} \end{array}\right], \qquad M_1 = 1\,,$$

$$C_k = \left[\begin{array}{c|c|c} C_{k-1} & 0 & C_{k-1} \\ \hline 0 & 1 & 1 \\ \hline C_{k-1} & 1 & \overline{C}_{k-1} \end{array}\right] \text{ e } \quad C_1 = 1. \tag{7}$$

Além disso, a matriz inversa C^{-1} é dada por

$$C_k^{-1} = \frac{2C_k - [\mathbf{1}]}{2^{k-1}}, \tag{8}$$

onde $[\mathbf{1}]$ é uma matriz com todos os elementos iguais a 1.

Exemplo 2 Considere um código de bloco 5-ário com $k = 1$. Então suas matrizes de pesos C_H (Hamming), C_L (Lee) e C_E (Euclidiana) são iguais a

$$C^* = \begin{bmatrix} 1 & 2 & 3 & 4 \\ 2 & 4 & 1 & 3 \\ 3 & 1 & 4 & 2 \\ 4 & 3 & 2 & 1 \end{bmatrix}, \quad C_H = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix},$$

$$C_L = \begin{bmatrix} 1 & 2 & 2 & 1 \\ 2 & 1 & 1 & 2 \\ 2 & 1 & 1 & 2 \\ 1 & 2 & 2 & 1 \end{bmatrix} \text{ e } \quad C_E = \begin{bmatrix} 1 & 4 & 9 & 16 \\ 4 & 16 & 1 & 9 \\ 9 & 1 & 16 & 4 \\ 16 & 9 & 4 & 1 \end{bmatrix}.$$

III. Construção de Códigos de Bloco com d_{min} Máxima

Usando a representação modular (1) podemos formular o problema de otimização linear inteira (9), cujas soluções fornecem o vetor N de um código de bloco de taxa $r = k/n$ com d_{min} máxima.

$$\begin{array}{ll} \text{Maximizar:} & z = w \\ \text{Sujeito a:} & N \cdot C \geq w[\mathbf{1}] \\ & \sum n_i = n \\ & n_i \geq 0, \text{ inteiros,} \end{array} \tag{9}$$

onde $[\mathbf{1}] = [1, 1, \ldots, 1]^T$ e w é igual a distância mínima.

Observe que caso N não fosse necessariamente composto por valores inteiros, então (9) poderia ser resolvido por exemplo pelo método simplex. Porém, N deve ser inteiro e neste caso existem pelo menos dois métodos apropriados para obter essas soluções, [6]. O primeiro método é o Plano de Corte (*Cutting Plane*), cuja idéia é adicionar restrições ao problema com o propósito de forçar a obtenção de uma solução inteira. A segunda técnica é baseada na divisão do problema original em uma série de pequenos problemas que são simples de resolver e então usar essa informação para resolver o problema inicial. Esse método é conhecido como *Branch and Bound*.

Contudo, observe que o código obtido como solução da formulação (9) pode não ser um código ótimo (melhor polinômio enumerador), pois garantimos apenas que esse código possui d_{min} máxima. Para garantir que iremos obter um código ótimo devemos adicionar outras restrições de otimalidade na resolução. Tais restrições serão detalhadas posteriomente para os métodos Plano de Corte e *Branch and Bound*.

Um recurso que demonstrou ser de grande utilidade para resolver (9) é uso de um limitante superior da distância mínima como mais uma restrição do problema. Verificamos que a adição dessa restrição faz com que a solução inteira seja encontrada mais rapidamente. Para o caso particular de códigos de bloco q-ários, $q = 2, \ldots, 9$, com distância de Hamming podemos usar o limitante de Brouwer [2]. Contudo, estamos trabalhando com uma medida de distância geral (f_d), portanto devemos determinar um novo limitante superior para essa distância. Para tanto, devemos mostrar inicialmente o seguinte resultado.

Teorema 1: A soma dos elementos de qualquer uma das linhas ou colunas da matriz de pesos, C, de um código de bloco sobre $\mathbb{F}_q$ de taxa $r = k/n$ é igual a

$$s = q^{k-1} \sum_{j=0}^{q-1} f_d(j) - f_d(0). \tag{10}$$

Demonstração: Sabemos que a matriz M possui como colunas todas as possíveis combinações de k elementos de $\mathbb{F}_q$, portanto cada linha de M possui $(q^{k-1}-1)$ valores 0's e q^{k-1} valores 1's, 2's até $q-1$. Como q é um número primo e as linhas da matriz C^* são combinações lineares das linhas de M sobre o corpo $\mathbb{F}_q$, então as linhas de C^* também possuem $(q^{k-1}-1)$ valores 0's e q^{k-1} valores 1's, 2's até $q-1$. Finalmente, se aplicarmos a função distância, $c_{ij} = f_d(c^*_{ij})$, e somarmos os elementos de qualquer linha de C obtemos o valor de s. ■

Usaremos esse resultado para calcularmos o seguinte limitante superior trivial para a distância mínima de um código de bloco.

Teorema 2 (Limitante Superior): A distância mínima de um código de bloco sobre $\mathbb{F}_q$ de taxa $r = k/n$ é limitada superiormente por

$$d_{min} \leq \frac{n}{q^k - 1} s = \frac{n}{q^k - 1} \left[q^{k-1} \sum_{j=0}^{q-1} f_d(j) - f_d(0) \right]. \tag{11}$$

Demonstração: Usando (3) e (10) temos que o somatório dos pesos das m palavras-códigos é igual a

$$\sum_{ij} n_i f_d(c^*_{ij}) = \sum_i n_i \sum_j f_d(c^*_{ji}) = s \sum_i n_i = n\, s.$$

Logo a distância mínima do código deve ser menor ou igual do que o somatório dos pesos de todas as palavras-códigos dividido pelo número total de palavras-código, $m = q^k - 1$. ■

Observe que caso a distância associada ao código seja a de Hamming, então temos de (4) e (11) que

$$d_{min} \leq \left\lfloor \frac{n(q-1)q^{k-1}}{q^k - 1} \right\rfloor, \tag{12}$$

onde $\lfloor a \rfloor$ é o maior inteiro menor ou igual a a. Por exemplo, $\lfloor 3.8 \rfloor = 3$ e $\lfloor -1.3 \rfloor = -2$. Observe que (12) coincide com o limitante superior de Plotkin para códigos q-ários. Caso q seja maior do que 2 e a medida de distância em questão seja a de Lee temos de (5) e novamente de (11) que

$$d_{min} \leq \left\lfloor \frac{n(q^2-1)q^{k-1}}{4(q^k - 1)} \right\rfloor. \tag{13}$$

Finalmente, se a distância for a euclidiana ao quadrado, temos de (6) e de (11), que

$$d_{min} \leq \left\lfloor \frac{n(q-1)(2q-1)q^k}{6(q^k - 1)} \right\rfloor. \tag{14}$$

Antes de descrevermos os métodos Plano de Corte e *Branch and Bound*, considere que dado um problema de otimização inteira (IP) como (9), existe um problema de otimização linear associado a ele chamado de relaxação linear (LR) formado pela alteração da restrição $n_i \in \mathbb{Z}_+$ para $n_i \in \mathbb{R}^+$. Assim, os seguintes resultados são imediatos:

1) O valor ótimo da função objetivo de (LR) é maior ou igual do que o valor de (IP).
2) Se (LR) não possui solução (*infeasible*), então (IP) também não possui.
3) Se os coeficientes da função objetivo forem inteiros, então o valor ótimo de (IP) é menor ou igual do que a parte inteira do valor ótimo de (LR).

A. Método Plano de Corte

Como dito anteriormente, o método plano de corte resolve o problema (9). A idéia deste método é adicionar novas restrições ao problema com o objetivo de forçar que a solução ótima do problema seja inteira. Uma restrição com tal finalidade é chamada de corte (*cut*). Um corte é gerado a partir de uma solução fracionária corrente da (LR) e possui o seguinte critério:

- toda solução inteira factível, é factível para o corte,
- a atual solução fracionária não é factível para o corte.

Uma maneira eficiente de gerar planos de corte é usar o corte de *Gomory*. Tal corte é obtido de uma das restrições gerada pelo método simplex para a solução corrente ótima da (LR), isto é, se tivermos a restrição

$$x_k + \sum a_i x_i = b_k,$$

sendo b_k um número não inteiro, então o corte para essa restrição é dada por

$$\sum (a_i - \lfloor a_i \rfloor) x_i \geq b_k - \lfloor b_k \rfloor. \tag{15}$$

Exemplo 3 Neste exemplo, vamos construir todos os códigos de bloco binários de taxa $r = 2/n$ com d_{min} máxima resolvendo o problema de otimização inteira (9) pelo método Plano de Corte. Para tanto, devemos inicialmente resolver o problema linear relaxado associado ao seguinte problema proveniente da formulação (9):

$$\begin{array}{lrl}
\text{Maximizar:} & & w \\
\text{Sujeito a:} & n_1 \quad\; + n_3 & \geq w \\
& n_2 + n_3 & \geq w \\
& n_1 + n_2 \quad\; & \geq w \\
& n_1 + n_2 + n_3 & = n \\
& n_1,\ n_2,\ n_3 & \geq 0,\ \text{inteiros.}
\end{array}$$

Portanto, se adicionarmos as variáveis de folga, que também devem possuir valores inteiros, temos a seguinte solução para

o problema linear relaxado

$$\begin{array}{lrl}
\text{Maximizar:} & & -\frac{1}{3}n_4-\frac{1}{3}n_5-\frac{1}{3}n_6+\frac{2}{3}n \\
\text{Sujeito a:} & n_1 & -\frac{1}{3}n_4+\frac{2}{3}n_5-\frac{1}{3}n_6=\frac{1}{3}n \\
& n_2 & +\frac{2}{3}n_4-\frac{1}{3}n_5-\frac{1}{3}n_6=\frac{1}{3}n \\
& n_3 & -\frac{1}{3}n_4-\frac{1}{3}n_5+\frac{2}{3}n_6=\frac{1}{3}n \\
& w & +\frac{1}{3}n_4+\frac{1}{3}n_5+\frac{1}{3}n_6=\frac{2}{3}n \\
& n_1,n_2,n_3,w, & n_4, \quad n_5, \quad n_6 \geq 0, \text{ inteiros.}
\end{array}$$

Neste caso, verifique que a solução ótima do problema linear, $x = [\frac{1}{3}n, \frac{1}{3}n, \frac{1}{3}n, \frac{2}{3}n, 0, 0, 0]$, é inteira se n for múltiplo de três. Então, podemos usar a quarta linha, associada a variável básica w fracionária, para gerar um novo corte

$$\frac{1}{3}n_4 + \frac{1}{3}n_5 + \frac{1}{3}n_6 \geq \frac{2}{3}n - \left\lfloor \frac{2}{3}n \right\rfloor,$$

Adicionando esse plano de corte e otimizando novamente o problema, obtemos

$$\begin{array}{lrl}
\text{Maximizar:} & & -s_1 + \left\lfloor \frac{2}{3}n \right\rfloor \\
\text{Sujeito a:} & n_1 \quad -n_4 & -n_6+2s_1 = 2\left\lfloor \frac{2}{3}n \right\rfloor - n \\
& n_2 \quad +n_4 & -s_1 = n - \left\lfloor \frac{2}{3}n \right\rfloor \\
& n_3 \quad +n_6 & -s_1 = n - \left\lfloor \frac{2}{3}n \right\rfloor \\
& w & +s_1 = \left\lfloor \frac{2}{3}n \right\rfloor \\
& & n_4+n_5+n_6-3s_1 = 2n - 3\left\lfloor \frac{2}{3}n \right\rfloor \\
& n_1,n_2,n_3,w, \; n_4, \; n_5, \; n_6, & s_1 \geq 0, \text{ inteiros.}
\end{array}$$

Observe que se $n > 1$ a solução ótima do problema é inteira. Com isso, podemos obter a representação modular para todos os códigos de bloco binários de taxa $r = 2/n$ com d_{min} máxima. Observe que os códigos com $n = 3i + 1$, possuem polinômio gerador $W_{\mathcal{C}} = 1 + 2t^{2i} + t^{2i+1}$. Porém, se transformarmos a solução básica do método simplex ($n_4 = n_6 = 0$, $n_5 = 2$) em uma solução não básica ($n_4 = n_6 = 1$, $n_5 = 0$), obtemos os códigos ótimos para $n = 3i + 1$ com $W_{\mathcal{C}} = 1 + t^{2i} + 2t^{2i+1}$. A Tabela I contém a representação modular de todos os códigos ótimos binários de taxa $r = 2/n$.

n	N	d_{min}	$W_{\mathcal{C}}$
$3i$	$[i, i, i]$	$2i$	$1 + 3t^{2i}$
$3i+1$	$[i, i, i+1]$	$2i$	$1 + t^{2i} + 2t^{2i+1}$
$3i+2$	$[i, i+1, i+1]$	$2i+1$	$1 + 2t^{2i+1} + t^{2i+2}$

TABELA I

CÓDIGOS DE BLOCO BINÁRIOS ÓTIMOS DE TAXA $r = 2/n$, $i = 1, 2, \ldots$.

Sabemos que a solução final fornecida pelo método Plano de Corte é uma solução básica do método simplex. Essa solução fornece a representação modular de códigos de bloco com d_{min} máxima, mas não podemos garantir que tais códigos sejam ótimos. Porém, podemos transformar convenientemente a solução básica do método simplex em uma solução não básica, de maneira que essa solução não básica possua polinômio enumerador melhor do que a solução básica, veja exemplo 3. Usamos esse procedimento para determinar os códigos binários ótimos com $r = 3/n$, veja Tabela II.

Usando o método Plano de Corte obtemos todos os códigos de bloco binários com d_{min} máxima e $k < 9$. Não trabalhamos com valores maiores de k pois o custo computacional

n	N	$W_{\mathcal{C}}$
$7i-3$	$4i-2$	$1 + 6t^{4i-2} + t^{4i}$
$7i-2$	$4i-2$	$1 + 2t^{4i-2} + 4t^{4i-1} + t^{4i}$
$7i-1$	$4i-1$	$1 + 4t^{4i-1} + 3t^{4i}$
$7i$	$4i$	$1 + 7t^{4i}$
$7i+1$	$4i$	$1 + 3t^{4i} + 4t^{4i+1}$
$7i+2$	$4i$	$1 + t^{4i} + 4t^{4i+1} + 2t^{4i+2}$
$7i+3$	$4i+1$	$1 + 3t^{4i+1} + 3t^{4i+2} + t^{4i+3}$

TABELA II

CÓDIGOS DE BLOCO BINÁRIOS ÓTIMOS DE TAXA $r = 3/n$, $i = 1, 2, \ldots$.

aumenta exponencialmente. Porém, o método abordado pode ser aplicado para qualquer valor de k, veja exemplo 4.

Exemplo 4 Neste exemplo vamos construir os códigos binários de Reed-Muller de primeira ordem, ou seja, códigos de taxa $r = k/2^{k-1}$ com $d_{min} = 2^{k-2}$, $k > 1$. Devemos ressaltar que esses códigos foram obtidos resolvendo o problema de otimização inteira (9) usando apenas um corte.

Primeiramente, vamos resolver o problema linear relaxado de (9), isto é,

$$\left[\begin{array}{c|c|c} I & 0 & \frac{2}{2^k-1} - 2^{2-k}C \\ \hline 0 & 1 & \frac{1}{2^k-1} \end{array}\right] \cdot \left[\begin{array}{c} N \\ \hline w \\ \hline N^+ \end{array}\right] = \left[\begin{array}{c} \frac{n}{2^k-1} \\ \hline \frac{n2^{k-1}}{2^k-1} \end{array}\right],$$

onde N^+ representa as $2^k - 1$ variáveis de folga. Observe que se n for múltiplo de $2^k - 1$, então a solução é inteira e código possui d_{min} múltipla de 2^{k-1}. Quando $n = 2^k - 1$ o código em questão é dual ao código de Hamming binário. Finalmente, se construirmos um corte proveniente da última equação,

$$n_1^+ + n_2^+ + \cdots = n2^{k-1} - (2^k - 1)\left\lfloor \frac{n2^{k-1}}{2^k - 1} \right\rfloor,$$

então teremos uma solução inteira para o novo problema linear relaxado se $n = 2^{k-1}$. Neste caso,

$$d_{min} = \left\lfloor \frac{2^{2k-2}}{2^k - 1} \right\rfloor = 2^{k-2}, \quad \text{se } k > 1,$$

e N é igual a qualquer uma das linhas da matriz C.

Ainda sobre o exemplo 4, podemos verificar que para cada corte construído temos mais uma classe de códigos gerada. Devemos ressaltar que esse fato é muito importante, principalmente, para a busca de novas classes de códigos de bloco com d_{min} máxima.

B. Método Branch and Bound

Como foi dito anteriormente, o método *Branch and Bound* divide o problema original em uma série de pequenos problemas que são simples de resolver e então usa essa informação para resolver o problema inicial.

Para entender melhor essa idéia considere o problema $z = \max\{cx : x \in S\}$. Assim, se $S = S_1 \cup \cdots \cup S_K$ for uma decomposição de S em conjuntos menores, e $z^k = \max\{cx : x \in S_k\}$ para $k = 1, \ldots, K$, então $z = \max_k\{z^k\}$.

A essência do algoritmo *Branch and Bound* pode ser descrita como segue:

1) Resolva o problema de relaxação linear, obtenha $\bar{x} = [\bar{x}_1, \ldots, \bar{x}_n]$.

2) Se a solução for inteira, então já temos a solução. Caso contrário, crie dois novos subproblemas de uma variável fracionária $\bar{x}_j$, isto é, $S_1 = S \cap \{x : x_j \leq \lfloor \bar{x}_j \rfloor\}$ e $S_2 = S \cap \{x : x_j \geq \lceil \bar{x}_j \rceil\}$.
3) Um subproblema não está ativo quando um dos seguintes fatos ocorre:
   - Todas as variáveis da solução são inteiras;
   - O subproblema não possui solução;
   - O subproblema pode ser eliminado por um limitante.
4) Escolha um subproblema ativo e crie dois novos subproblemas de uma variável fracionária.
5) Repita até que não existam mais subproblemas ativos.

Exemplo 5 Neste exemplo vamos construir um código de bloco binário de taxa $r = 3/5$ com d_{min} máxima usando a formulação (9) e o método *Branch and Bound*, veja Figura 1.

$$
\begin{aligned}
\text{Maximizar:} \quad & z = w \\
\text{Sujeito a:} \quad & n_1 + n_3 + n_5 + n_7 \geq w \\
& n_2 + n_3 + n_6 + n_7 \geq w \\
& n_1 + n_2 + n_5 + n_6 \geq w \\
& n_4 + n_5 + n_6 + n_7 \geq w \\
& n_1 + n_3 + n_4 + n_6 \geq w \\
& n_2 + n_3 + n_4 + n_5 \geq w \\
& n_1 + n_2 + n_4 + n_7 \geq w \\
& n_1 + n_2 + \cdots + n_7 = 5 \\
& n_i \geq 0, \text{ inteiros.}
\end{aligned}
$$

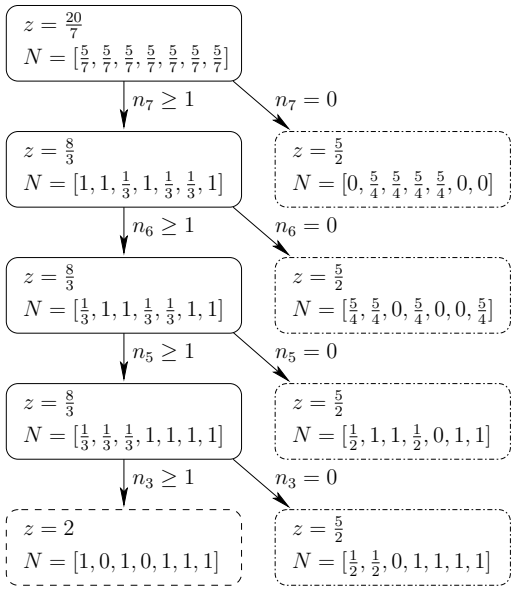

Fig. 1. Construção de um código de bloco binário de taxa $r = 3/5$ usando o método *Branch and Bound*.

Resolvendo o primeiro passo do método *Branch and Bound* encontramos a seguinte solução para o problema linear relaxado: $N = [\frac{5}{7}, \ldots, \frac{5}{7}]$ e $z = \frac{20}{7}$. Note que a solução inteira não pode ser maior do que $\lfloor \frac{20}{7} \rfloor = 2$.

Segundo o algoritmo devemos dividir o problema original em dois subproblemas, pois a solução não é inteira. Neste caso, podemos tomar, por exemplo, um subproblema no qual $n_7 \geq 1$ e outro no qual $n_7 = 0$. Observe que ambos os subproblemas estão ativos, então vamos sempre escolher o subproblema ativo com função objetivo de maior valor para dividir, no caso o subproblema com $n_7 \geq 1$.

Repetindo esse procedimento mais três vezes obtemos um código de bloco binário e taxa $r = 3/5$ e $d_{min} = 2$. Contudo, note da Figura 1 que ainda restaram subproblemas ativos (borda com traço-ponto), cujas funções objetivos possuem valo r$\frac{5}{2}$. Portanto, se desenvolvermos esses subproblemas os códigos obtidos possuirão d_{min} de no máximo 2.

Como mencionado anteriormente, o código obtido como solução da formulação (9) pode não ser um código ótimo, pois garantimos apenas que esse código possui d_{min} máxima. Neste caso, para obtermos um código ótimo é necessário, mas não suficiente, que na resolução do *Branch and Bound* optemos sempre por desenvolver os subproblemas com melhor solução lexicográfica. Para que essa condição torne-se suficiente devemos desenvolver todas as restrições ativas que possam fornecer o melhor dos polinômios enumeradores.

Contudo, essa modificação do algoritmo pode aumentar o custo computacional do *Branch and Bound*, pois devemos desenvolver mais restrições ativas e substituir o método simplex, usado para resolver cada subproblema, por um algoritmo que forneça a solução lexicográfica ótima do subproblema.

IV. Simplificações do Problema de Otimização

Nesta seção vamos apresentar algumas simplificações úteis que podem ser aplicadas na solução do problema (9) com o objetivo de propiciar a criação de novas classes de códigos de bloco sobre $\mathbb{F}_q$ e minimizar o esforço computacional dos métodos de otimização descritos.

Caso a medida de distância de um código de bloco sobre $\mathbb{F}_q$ seja a de Hamming, então podemos obter o seguinte resultado.

Teorema 3: A matriz de pesos C de um código de bloco sobre $\mathbb{F}_q$, com k entradas e distância de Hamming (4) possui apenas $(q^k - 1)/(q - 1)$ linhas ou colunas diferentes.

Demonstração: Verifique que para cada coluna c_i da matriz M existem $q - 2$ outras colunas dadas por $a \cdot c_i$, $a \in \{2, 3, \ldots, q - 1\}$. Portanto, $C^* = (M' \cdot M) \bmod q$ também possui linhas ou colunas com esta propriedade. Logo, usando a definição (4) e o fato de $a \neq 0$, verificamos que cada linha de C repete $q - 1$ vezes. ■

Exemplo 6 Neste exemplo vamos usar o Teorema 3 para simplificar a construção de códigos ternários com $r = 2/n$ e d_{min} máxima. De (9) podemos obter a seguinte formulação

$$
\begin{aligned}
&\text{Maximizar: } z = w \\
&\text{Sujeito a:} \\
&\begin{bmatrix} 1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 0 & 1 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} n_1 \\ n_2 \\ n_3 \\ n_4 \\ n_5 \\ n_6 \\ n_7 \\ n_8 \end{bmatrix} \geq \begin{bmatrix} w \\ w \\ w \\ w \\ w \\ w \\ w \\ w \end{bmatrix} \\
&n_1 + n_2 + n_3 + n_4 + n_5 + n_6 + n_7 + n_8 = n \\
&n_i \geq 0, \text{ inteiros.}
\end{aligned}
$$

Note que as linhas de C possuem a seguinte característica: $l_1 = l_2$, $l_3 = l_6$, $l_4 = l_8$ e $l_5 = l_7$. Portanto, podemos simplificar o problema de otimização para

Maximizar: $z = w$
Sujeito a:

$$\begin{bmatrix} 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} n_1 \\ n_3 \\ n_4 \\ n_5 \end{bmatrix} \geq \begin{bmatrix} w \\ w \\ w \\ w \end{bmatrix}$$

$$n_1 + n_3 + n_4 + n_5 = n$$
$$n_2 = n_6 = n_7 = n_8 = 0$$
$$n_i \geq 0, \text{ inteiros.}$$

Então, usando alguma técnica de otimização inteira podemos encontrar os códigos de bloco ternários com d_{min} máxima e taxa $r = 2/n$, veja Tabela III.

n	N	$W_{\mathcal{C}}$
$4i-1$	$[i, 0, i, i, i-1, 0, 0, 0]$	$1 + 6t^{3i-1} + 2t^{3i}$
$4i$	$[i, 0, i, i, i, 0, 0, 0]$	$1 + 8t^{3i}$
$4i+1$	$[i+1, 0, i, i, i, 0, 0, 0]$	$1 + 2t^{3i} + 6t^{3i+1}$
$4i+2$	$[i+1, 0, i+1, i, i, 0, 0, 0]$	$1 + 4t^{3i+1} + 4t^{3i+2}$

TABELA III
CÓDIGOS DE BLOCO TERNÁRIOS DE TAXA $r = 2/n$ COM d_{min} MÁXIMA.

A simplificação usada no exemplo anterior pode ser aproveitada para gerar uma classe de códigos de bloco sobre $\mathbb{F}_q$ com $n = i(q^k - 1)/(q-1)$ e $d_{min} = iq^{k-1}$, máxima. Quando $i = 1$, essa classe de códigos é dual à dos códigos de Hamming q-ários, q primo.

O próximo resultado é uma conjectura útil na construção de classes de códigos de bloco sobre $\mathbb{F}_q$ com d_{min} máxima.

Conjectura: Se um código de bloco $\mathcal{C}_0$ sobre $\mathbb{F}_q$ com taxa $r = k/n_0$ e representação modular N_0 tiver $d_{min} = d_0$ máxima, então o código de bloco $\mathcal{C}_+$ sobre $\mathbb{F}_q$ com representação modular $N = N_0 + j[\mathbf{1}]$, $j = 1, 2, \ldots$, possui taxa $r = k/(n_0 + j(q^k - 1))$ e $d_{min} = d_0 + js$ também máxima, onde s é dado por (10).

Analisando a conjectura anterior, não é difícil verificar que as diferenças entre as d_{min} de $\mathcal{C}_0$ e $\mathcal{C}_+$ e seus respectivos limitantes superiores (11) são iguais, logo se a d_{min} de $\mathcal{C}_0$ atinge seu limitante então a d_{min} de $\mathcal{C}_+$ também atinge seu limitante. Portanto, a conjectura deve ser provada também para os códigos $\mathcal{C}_o$ cujas d_{min} não atingem o limitante superior.

Como conseqüência imediata da conjectura anterior temos que é suficiente conhecermos apenas os códigos de bloco sobre $\mathbb{F}_q$ com d_{min} máxima de $r = k/i$, $i = k+1, \ldots, q^k + k$, para obtermos todos os códigos com as mesmas k entradas e d_{min} máxima.

Com base nos resultados já apresentados, verificamos que uma possível simplificação da formulação (9) é o fato da representação modular de N satisfazer

$$\max_{i,j}\{|n_i - n_j|\} \leq 1\,,$$

para quaisquer valores de n_i e n_j.

Caso essa simplificação seja realmente possível, então podemos resolver (9) usando, por exemplo, o método *Branch and Bound* substituindo as restrições de desigualdade $x_j \leq \lfloor \bar{x}_j \rfloor$ e $x_j \geq \lceil \bar{x}_j \rceil$ por restrições de igualdade. Além disso, quando usamos o método Plano de Corte essa restrição faz com que o número de cortes seja significativamente reduzido.

Devemos ressaltar que uma lista contendo possíveis novos códigos blocos está sendo construída utilizando os métodos propostos neste trabalho.

V. CONCLUSÕES

A generalização da representação modular para códigos de bloco sobre $\mathbb{F}_q$ com uma medida de distância geral demonstra ser coerente e útil. Sua aplicação na formulação (9) propiciou a construção de códigos de blocos já conhecidos e pode possibilitar a construção de novas classes de códigos, principalmente, as que não possuem distância de Hamming. Quanto aos métodos de otimização apresentados, verificamos que o Plano de Corte é mais adequado para a construção de classes de códigos, enquanto que o método *Branch and Bound* aparenta ser mais eficiente na determinação de um código por vez. As simplificações do problema de otimização proposto são úteis para a criação de classes de códigos e na minimização do esforço computacional. Contudo, a construção de determinados códigos de bloco sobre $\mathbb{F}_q$ com muitas entradas (k grande) pode se tornar impraticável.

AGRADECIMENTOS

Os autores agradecem o CNPq, a FAPESP e a CAPES pelo apoio financeiro no período de desenvolvimento desse trabalho.

REFERÊNCIAS

[1] A. E. Brouwer, "The Linear programming bound forbinary linear codes," *IEEE Trans. Inform. Theory*, vol. 39, pp. 677-680, Mar. 1993.

[2] A. E. Brouwer, Linear Code Bounds. [Online]. Avaliable: http://www.win.tue.nl/∼aeb/voorlincod.html

[3] A. Said, and R. Palazzo, Jr., "Using combinatorial optimization to design good unit-memory convolutional codes," *IEEE Trans. Inform. Theory*, vol.IT-39, pp. 1100-1108, May 1993.

[4] G. C. Clark and J. B. Cain, *Error Correction Coding for Digitar Communications*. New York: Plenum, 1988.

[5] G. Ferrari and K. M. Chugg, "Linear programming-based optimization of the distance spectrum of linear block codes," *IEEE Trans. Inform. Theory*, vol. 49, pp. 1794-1800, July 2003.

[6] L. A. Wolsey, *Integer Programming*, John Wileyand Sons, New York, 1998.

[7] R.G. Cavalcante, e R. Palazzo Jr., "Análise de desempenho de constelações de sinais geometricamente uniformes provenientes de reticulados $\{p, q\}$ em espaços bidimensionais com curvatura constante," *XXI Simpósio Brasileiro de Telecomunicações-SBrT'04*, Belém, 2004.

[8] R.G. Cavalcante, H. Lazari, J.D. Lima, and R. Palazzo Jr., "A new approach to the design of digital communication systems,"*AMS-DIMACS Series*, pp.1-32, August 2005.

[9] R. Palazzo, Jr., "A network flow approach to convolutional codes," *IEEE Trans. on Comm.*, vol.COM-41, pp. 1429-1440, April 1995.

[10] W. W. Peterson and E. J. Weldon, Jr., *Error Correcting Codes*, 2nd. ed., MIT Press, Cambridge, Mass, 1972.