

Limites Superiores para o Problema de Código de Cobertura Mínima via Busca Tabu Reativa

Mayra C. Albuquerque
Marcus Poggi

Departamento de Informática
PUC-Rio, Rio de Janeiro–RJ, Brasil
{malbuquerque, poggi}@inf.puc-rio.br

André G. Castoldi
Emerson L. do Monte Carmelo

Departamento de Matemática
Universidade Estadual de Maringá, Maringá-PR, Brasil
guerinocastoldi@yahoo.com.br, elmcarmelo@gmail.com

Resumo

Neste trabalho, abordamos o Problema de Código de Cobertura Mínima no espaço de *Rosenbloom - Tsfasman* (RT), o qual estende o problema clássico de cobertura no espaço de *Hamming*. Considera-se $K_q^{RT}(m, s, R)$ a cardinalidade mínima da cobertura de código no RT-espaço induzida pela métrica *Rosenbloom - Tsfasman*. Uma reformulação do Problema de Código de Cobertura Mínima é apresentada em termos de conjunto dominante em grafos. Com esta nova reformulação, propõe-se uma meta-heurística Busca Tabu Reativa a fim de obter novos limites superiores para $K_q^{RT}(m, s, R)$.

Palavras-Chave: Problema de Código de Cobertura, Métrica Rosenbloom-Tsfasman, Busca Tabu Reativa.

Abstract

In this work, we investigate the Covering Codes Problem in the *Rosenbloom-Tsfasman* (RT) spaces, which extends the classical Covering Problem in Hamming Space. Let be $K_q^{RT}(m, s, R)$ the minimal cardinality of covering codes in the RT-space induced by the metric RT. A reformulation of the Problem Covering Codes is presented in terms of dominating set in graphs. From this reformulation, a Reactive Tabu Search metaheuristic is proposed in order to obtain new upper bounds for $K_q^{RT}(m, s, R)$.

Keywords: Covering Codes Problem, Rosenbloom-Tsfasman metric, Reactive Tabu Search.

1 Introdução

O problema de cobertura de códigos no RT-espaço é induzido pela métrica *Rosenbloom - Tsfasman*. Este problema é uma extensão do problema clássico de cobertura no espaço de *Hamming*. O problema de determinar a cardinalidade mínima $K_q(n, R)$ de uma cobertura no espaço de Hamming foi apresentado por Taussky e Todd (1948) e inicialmente investigado no caso que $R = 1$. Carnielli (1985) generalizou o problema de coberturas em espaços de Hamming para $1 < R < n$.

Dados $x, y \in \mathbb{Z}_q^n$, a distância de Hamming é definida por

$$d_H(x, y) = |\{i \in \{0, \dots, n\} : x_i \neq y_i\}|.$$

O problema de cobertura em espaços de Hamming consiste em determinar um subconjunto C de \mathbb{Z}_q^n de cardinalidade mínima tal que para todo $x \in \mathbb{Z}_q^n$, existe $c \in C$ satisfazendo $d_H(x, c) \leq R$.

Rosenbloom e Tsfasman (1997) definiram uma métrica sobre o espaço das matrizes $M_{m \times s}(\mathbb{F}_q)$, onde \mathbb{F}_q denota um corpo finito. Tal métrica generaliza a métrica de Hamming e algumas aplicações teóricas para esta métrica foram apresentadas por Rosenbloom e Tsfasman (1997). Esta métrica ficou conhecida como a métrica de Rosenbloom-Tsfasman, geralmente abreviada como RT-métrica ou RT-distância, e o espaço munido com esta métrica como RT-espaço.

Recentemente, há um grande interesse em códigos relacionados a esta métrica. Os seguintes assuntos foram investigados com a RT-métrica: códigos lineares (Ozen e Siap, 2006), as identidades de MacWilliams (Dougherty e Skriganov, 2002b, Ozen e Siap, 2004), códigos MDS (Dougherty e Skriganov, 2002a), o problema de empacotamento (Quistorff, 2007), entre outros.

Antes de definir a RT-métrica é preciso definir uma métrica auxiliar. Seja o espaço s -dimensional \mathbb{Z}_q^s . Dados $x = (x_1, \dots, x_s)$ e $y = (y_1, \dots, y_s)$ em \mathbb{Z}_q^s considere a métrica d_s definida por

$$d_s(x, y) = \max\{i \in \{0, 1, \dots, s\} : x_i \neq y_i\}.$$

A partir desta métrica, RT-métrica é definida, porém de uma maneira diferente daquela apresentada em Rosenbloom e Tsfasman (1997). Dado $x \in \mathbb{Z}_q^{ms}$ pode-se escrever

$$x = (x_{11}, \dots, x_{1s}, \dots, x_{j1}, \dots, x_{js}, \dots, x_{m1}, \dots, x_{ms}).$$

A distância de Rosenbloom e Tsfasman (ou *RT-distância*) entre dois vetores x e y de \mathbb{Z}_q^{ms} é definida como

$$d_{RT}(x, y) = \sum_{j=1}^m d_s((x_{j1}, \dots, x_{js}), (y_{j1}, \dots, y_{js})).$$

Em particular, se $s = 1$ tem-se a distância de Hamming.

Considere $(0, 1, 1, 0, 0, 1)$ e $(0, 0, 0, 0, 1, 1)$ dois vetores em \mathbb{Z}_2^6 . Seja $s = 2$ e $m = 3$, então: $d_{RT}((0, 1, 1, 0, 0, 1), (0, 0, 0, 0, 1, 1)) = 3 + 2 = 5$. Considerando, ainda, os vetores anteriores e $s = 3$, $m = 2$, segue que: $d_{RT}((0, 1, 1, 0, 0, 1), (0, 0, 0, 0, 1, 1)) = 2 + 1 + 1 = 4$. Porém, a distância de Hamming é: $d_H((0, 1, 1, 0, 0, 1), (0, 0, 0, 0, 1, 1)) = 3$. Em geral, vale que $d_H(x, y) \leq d_{RT}(x, y)$.

Castoldi e Carmelo (2013) generalizaram o problema de cobertura para o RT-espaço da seguinte maneira. Um código C é uma R -cobertura do RT-espaço \mathbb{Z}_q^{ms} , se para todo $x \in \mathbb{Z}_q^{ms}$

existe um vetor $c \in C$, tal que $d_{RT}(x, c) \leq R$. Dados os inteiros $q \geq 2, m \geq 1, s \geq 1$ e $R \geq 0$, denota-se $K_q^{RT}(m, s, R)$ como a cardinalidade mínima de uma R -cobertura do RT-espaço \mathbb{Z}_q^{ms} . Os códigos de cobertura no RT-espaço não foram explorados de maneira geral. A única contribuição sobre esse assunto foi apresentado por Yildiz et al. (2010), onde apresenta o problema de cobertura utilizando a métrica d_s , que é um caso particular da RT-métrica. Neste caso, o valor exato para o código de cobertura é apresentado.

O principal objetivo do trabalho é encontrar melhores limites superiores para o código de \mathbb{Z}_q^{ms} por meio de uma variação da meta-heurística Busca Tabu, a Busca Tabu Reativa. O presente artigo está estruturado da seguinte maneira. Alguns resultados preliminares serão apresentados na Seção 2. A Seção 3 descreve a reformulação do problema de cobertura em termos de grafos dominantes. A Seção 4 discute a meta-heurística Busca Tabu Reativa e os detalhes para a sua implementação. Finalmente, na Seção 5, os resultados obtidos e a conclusão são apresentados.

2 Limites preliminares

Nesta seção, os limites triviais e duas classes exatas de $K_q^{RT}(m, s, R)$ são apresentados. Além disso, a existência de uma relação entre as funções $K_q(n, R)$ e $K_q^{RT}(m, s, R)$ será demonstrada. Salvo menção em contrário, reporta-se nesta seção alguns resultados em Castoldi (2012).

Seja $V_q^{RT}(m, s, R)$ a cardinalidade de uma RT-bola de raio R no RT-espaço \mathbb{Z}_q^{ms} . Os limites triviais são apresentados abaixo.

Proposição 1 Para cada $m, s, q \geq 2$ e R tal que $0 < R < ms$,

1. $K_q^{RT}(m, s, R) \geq \frac{q^{ms}}{V_q^{RT}(m, s, R)}$.
2. $K_q^{RT}(m, s, R) \leq q^{ms-R}$.

A relação a seguir mostra que $K_q(ms, R)$ é um limite inferior para $K_q^{RT}(m, s, R)$.

Proposição 2 Para todo $q \geq 2$ e $0 < R < ms$,

$$K_q(ms, R) \leq K_q^{RT}(m, s, R).$$

O valor exato a seguir foi provado por Yildiz et al. (2010).

Teorema 1 Para $m = 1$ e para todo $s, q \geq 2$,

$$K_q^{RT}(1, s, R) = q^{s-R}.$$

A generalização de um resultado obtido por Carnielli (1985) é apresentado.

Teorema 2 Se $m \geq (t - 1)q + 1$, com $q \geq 2$ e $t \geq 1$, então

$$K_q^{RT}(m, s, ms - t) = q.$$

3 Códigos de Cobertura e Conjunto Dominante em Grafos

Nesta seção apresentamos uma reformulação de $K_q^{RT}(m, s, R)$ por meio de conjunto dominante em grafos.

Considere um grafo não-orientado $G = (V, E)$, em que V é o conjunto de nós e E o conjunto de arestas. Seja U um subconjunto de nós de V ($U \subseteq V$), dizemos U é um conjunto dominante do grafo G , se para cada nó $v \in V$ tem-se que v está em U ou existe um nó $u \in U$ tal que u é adjacente ao nó v . O objetivo do Problema do Conjunto Dominante é encontrar um conjunto dominante U de tamanho mínimo no grafo G . Este problema é NP-difícil, como demonstrado em (Garey e Johnson, 1979).

Seja \mathbb{Z}_q^{ms} o conjunto de todos os vetores no espaço ms -dimensional com componentes em $\{0, 1, \dots, q - 1\}$ e seja R um inteiro tal que $0 < R < ms$.

Um grafo $G(V, E) = G(m, s, q, R)$ pode ser construído como segue. Dados q, m, s e R , o grafo em questão considera que $V = \mathbb{Z}_q^{ms}$, assim, cada vetor $u \in \mathbb{Z}_q^{ms}$ está associado a um vértice de V . Com isso, o número de vértices do grafo G será $|V| = q^{ms}$. As arestas $e = uv$ do grafo G serão definidas de acordo com a cardinalidade dada pela métrica *Rosenbloom - Tsfasman (RT)*. A distância RT entre u e v é denotada por $d_{RT}(v, u)$. Desta forma, a aresta e está em E se, e somente se, $0 < d_{RT}(v, u) \leq R$.

A construção acima é ilustrada pelo seguinte exemplo. Considere o problema de 2-cobertura no RT -espaço \mathbb{Z}_2^4 . A Figura 1 representa o grafo $G(2, 2, 2, 2)$ referente a 2-cobertura. Note que, o número de vértices do grafo G é $q^{ms} = 2^{2 \cdot 2} = 16$.

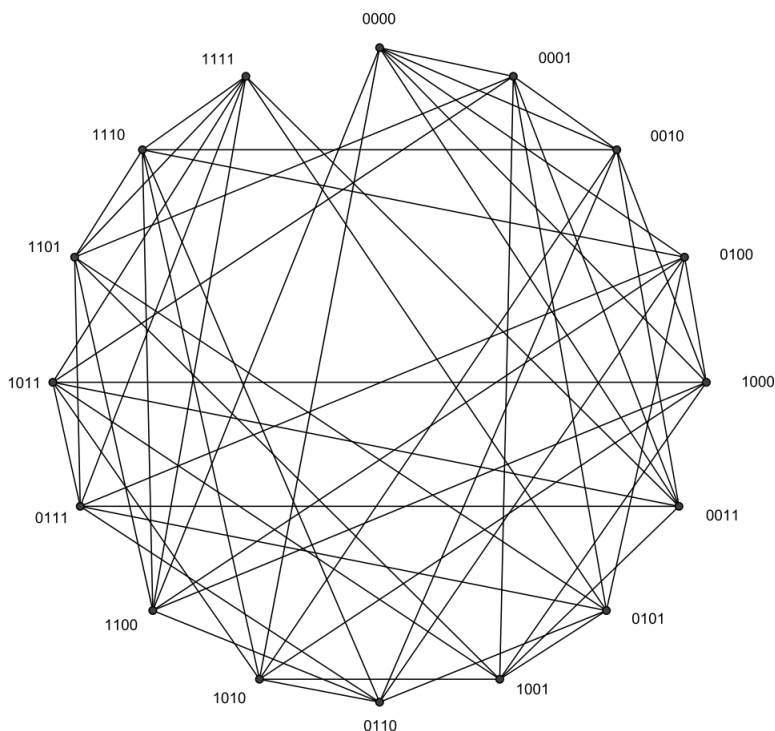


Figura 1: Grafo correspondente à cobertura $K_2^{RT}(2, 2, 2)$.

Observe que o conjunto dominante de $G(m, s, q, R)$ corresponde a uma cobertura de códigos em \mathbb{Z}_q^{ms} . Portanto, uma cobertura mínima em \mathbb{Z}_q^{ms} pode ser encontrada resolvendo o problema de conjunto dominante no grafo $G(m, s, q, R)$.

4 Meta-Heurística Busca Tabu Reativa

Carnielli et al. (1995), Mendes et al. (2010) apresentam o uso de meta-heurísticas para encontrar limites superiores para o código de R -cobertura no espaço de Hamming. Estes trabalhos apresentam bons limites para a R -cobertura no espaço de Hamming. O presente artigo propõe uma meta-heurística para o problema de cobertura no RT-espaço, mais especificamente, a Busca Tabu Reativa. Esta seção apresenta como essa abordagem foi feita.

A meta-heurística Busca Tabu (BT) clássica, apresentada em (Glover e Laguna, 1997) é baseada na exploração contínua no espaço de busca. A BT realiza um procedimento adaptativo fazendo uma busca local, em que a melhor solução na vizinhança substituirá a solução atual. O principal objetivo da BT é evitar a ciclagem que pode ocorrer ao intensificar a busca em regiões no espaço ditas promissoras.

A BT consiste em explorar o espaço de soluções movendo-se de uma solução para outra que seja seu melhor vizinho. Para realizar esse movimento é necessário definir a vizinhança de uma solução. O movimento nessa vizinhança é um passo guloso na maioria das vezes. Entretanto, só será permitido mover-se da solução atual para a vizinha, se o movimento necessário para isto não esteja classificado com ‘tabu’, i.e., o movimento não está proibido.

O esquema clássico da busca tabu mantém uma lista de movimentos classificados como tabu, chamada lista tabu. Na BT clássica, a lista tabu possui um tamanho fixo, o que permite ciclos na busca de tamanho maior que o tamanho da mesma. Embora este esquema possa levar a excelentes resultados, ainda é possível que a busca fique presa em um mínimo local. A fim de evitar a ciclagem, um mecanismo de reação na busca tabu é apresentado em Battiti e Tecchiolli (1994).

Ao incorporar o mecanismo de reação à estrutura básica da BT temos a Busca Tabu Reativa (BTR). A ideia é combinar os principais elementos da BT através de uma memória de longa duração que mantém o histórico das soluções visitadas durante a busca. Esta memória é usada para aumentar o tamanho da lista tabu quando há ciclos na busca.

Além de controlar dinamicamente o tamanho da lista tabu, a BTR utiliza outro mecanismo para escapar de mínimos locais, a perturbação. As perturbações são feitas de maneiras sucessivas e aleatórias até que o número de soluções visitadas supere um determinado limiar. A BTR vem apresentando sucesso em diversas aplicações como demonstrado nos trabalhos de Nanry e Barnes (2000), Osman e Wassan (2002), Bastos e Ribeiro (2002).

4.1 O esquema básico da Busca Tabu

O esquema da BT se inicia pela definição do espaço de busca. No problema de cobertura consideramos o espaço como sendo o conjunto de todos $2^{q^{ms}}$ subconjuntos de \mathbb{Z}_q^{ms} , que é o conjunto de vértices de $G(m, s, q, R)$. Como consequência nem todos os elementos deste espaço formam um conjunto dominante, isto é, um subconjunto de V não necessariamente é uma solução viável para o problema original. Esta escolha implica em trabalhar em um espaço no qual se tem uma estrutura de vizinhança. Contudo uma desvantagem é a necessidade de utilização de um mecanismo de penalidade. Esse mecanismo visa explorar, de forma frequente, soluções viáveis, ou seja, os conjuntos dominantes.

Se $U \subseteq V$ é uma solução atual, ou seja, uma cobertura viável, então a vizinhança de U é dada pelo conjunto de todas as soluções obtidas por U da seguinte forma: pela remoção de um vértice de U ou adição de um vértice de $V \setminus U$ a U . Isto leva a uma vizinhança de tamanho $|V| = q^{ms}$, o que em alguns casos, pode ser uma vizinhança grande. A função objetivo também possui um componente de penalidade. Lembrando que, a função objetivo é o tamanho do conjunto dominante, segue que a penalidade será o produto de um fator de

penalidade com o número de vértices que não foram cobertos pela solução atual. Quanto maior é o fator de penalidade maior a chance de obtermos soluções viáveis durante a busca. Porém, se o fator de penalidade é muito alto a busca se comporta como se apenas soluções viáveis fossem permitidas, tornando assim uma busca muito restritiva. Com isto, é preciso ajustar este fator de penalidade, tarefa importante na calibração da busca tabu. Isto se deve ao fato de que, o fator de penalidade balanceia o grau de liberdade da busca com a necessidade de obter soluções viáveis.

O fator de penalidade (ALPHA) é iniciado com um pequeno valor (MIN_ALPHA). A cada iteração é aumentado por um valor constante (STEP_ALPHA), até que o mesmo atinja um valor máximo (MAX_ALPHA). Atingindo o valor máximo, seu valor retorna ao inicial (MIN_ALPHA). Com esta estratégia é possível que a busca caminhe com grau de liberdade maior no início de cada ciclo e assim, venha a convergir gradualmente para uma solução viável.

A lista tabu armazena a última iteração na qual um movimento foi tornado tabu. O movimento pode ser adição ou remoção de um vértice. Desta forma, dois valores por vértices de V são guardados na lista: a última iteração em que foi adicionada a solução e a última iteração em que foi removida da mesma, contemplando todos os movimentos possíveis. Um movimento m é tabu se a diferença entre seu valor na lista tabu e o número da iteração atual é maior que o tamanho da lista tabu, usualmente referenciado por *tabu tenure*. Este valor do tabu tenure será objeto do mecanismo de reação, descrito mais adiante.

Outro elemento importante da busca tabu é o critério de aspiração. Este critério retira o status tabu de um movimento. O critério de aspiração utilizado revoga a restrição tabu de um movimento quando o mesmo leva a uma solução cujo valor da função objetivo é o melhor encontrado durante toda a busca.

O pseudo-código do esquema básico da BT para o problema de cobertura é apresentado pelo Algoritmo 1.

O algoritmo inicia construindo uma solução viável U , isto é, um conjunto dominante de $G(m, s, q, R)$ (linha 2). A partir de então, os elementos da busca tabu são inicializados. O loop principal, linha 7, é repetido um número suficientemente grande de vezes usualmente dado pelo tempo disponível de CPU. Até que não atinja a iteração máxima, o fator de penalidade é ajustado, linha 8. Logo após, o critério tabu dos movimentos e o critério de aspiração são verificados (linha 9). A linha 13 atualiza a melhor solução encontrada e, em seguida, na linha 15, o mecanismo de reação executado, este mecanismo é descrito a seguir.

4.2 O mecanismo de reação

O controle do tabu tenure e uma eventual mudança aleatória na solução atual são os principais elementos do mecanismo de reação. A ideia principal é manter uma memória a longo prazo que mantém o conjunto das soluções visitadas, a iteração da última visita e o número de visitas às mesmas. O objetivo desta memória é detectar a ciclagem entre as soluções. Cada vez que uma solução se repete na busca o intervalo entre as visitas é então calculado. A chamada reação rápida ocorre quando este intervalo é menor que um dado limiar (*'threshold'*). A reação consiste em atribuir ao tabu tenure um valor grande. A chamada reação lenta segue gradualmente reduzindo o valor do tabu tenure, já que se espera atingir um nova região do espaço de busca ao atribuir ao tabu tenure um valor alto.

Além deste controle dinâmico do valor do tabu tenure, existe outro mecanismo de escape de mínimos locais. Este mecanismo trabalha monitorando as soluções que são visitadas um número excessivo de vezes. Quando o número destas soluções excede um determinado limiar,

Algoritmo 1: Busca Tabu

Entrada: Grafo $G(m, s, q, R)$ e parâmetros
Saída: Vetor de solução $BestCode$

```

1 início
2   Construa uma cobertura inicial  $U$ , adicionando palavras aleatórias que não está
   em  $U$ , até que  $U$  seja uma cobertura;
3    $BestCode \leftarrow U$ ;
4    $T[m] \leftarrow 0$ , para cada movimento possível  $m$ . ;           // Atualiza a lista tabu
5    $ALPHA \leftarrow MIN\_APLHA$ . ;                               // Atualiza o fator de penalidade
6   Inicializa o mecanismo de reação.
7   enquanto  $iter \leq iterMax$  faça
8     Atualiza ALPHA;
9     Seleciona o movimento  $m$  que não é tabu ou que satisfaça o critério de
   aspiração.
10    Executa o movimento  $m$  em  $U$ .
11     $T[m] \leftarrow iteracao - atual$ ; ;                       // Atualiza o valor tabu de  $m$  em  $T$ 
12    se  $U$  é uma cobertura  $|U| < |BestCode|$  então
13      |  $BestCode \leftarrow U$ .
14    fim
15    Executa o mecanismo de reação.
16  fim
17 fim

```

um determinado número de movimentos aleatórios é realizado na solução atual, na esperança de quem a busca atinja um região do espaço ainda não explorada.

A combinação destes dois mecanismos torna a busca reativa uma importante melhoria no busca tabu clássica, melhorando seus resultados em diversas aplicações da busca tabu.

Para implementação foi necessário o uso de algumas estruturas descritas a seguir. A inicialização do mecanismo de reação apenas cria e inicializa estas variáveis e conjuntos.

- i) *Visited*: conjunto que mantém as informações das soluções visitadas durante a busca;
- ii) *OftenRepeated*: conjunto que armazena as soluções visitadas um número excessivo de vezes, sendo sua cardinalidade mantida na variável *chaotic*;
- iii) e as variáveis: *countLastSizeChange* que mantém o número de iterações desde a última mudança no tabu tenure, e *movingAvg* que armazena o tamanho médio dos ciclos completados durante a busca.

O pseudo-código do mecanismo de reação proposto é apresentando no Algoritmo 2.

O mecanismo de reação é executado após cada movimento na solução atual. Primeiramente o mecanismo testa se a solução atual já foi visitada anteriormente. Caso isso ocorra, o intervalo entre as visitas é então calculado e comparado ao valor de *CYCLE_MIN*, se for menor que este valor, então o mecanismo de reação incrementa o valor do tabu tenure através da multiplicação do mesmo pela constante *INCREASE* (*INCREASE* > 1). A partir daí, atualiza a variável *movingAVG* e o conjunto *OftenRepeated*. Quando o número de visitas a esta solução excede a constante *MAX_VISITS* e a solução ainda está no conjunto *OftenRepeated* então a solução será inserida neste conjunto. O último teste verifica se a

Algoritmo 2: Busca Tabu Reativa - Mecanismo de Reação

Entrada: Solução atual U
Saída: Vetor de solução $BestCode$

```

1 início
2    $Visited \leftarrow \emptyset$ ;
3    $OftenRepeated \leftarrow \emptyset$ ;
4    $chaotic \leftarrow 0$ ;
5    $countLastSizeChange \leftarrow 0$ ;
6    $movingAvg \leftarrow 0$ ;
7    $tabuTenure \leftarrow MIN\_tenure$ ;
   // Início do mecanismo de reação
8    $escape \leftarrow TRUE$ ;
9    $countLastSizeChange ++$ ;
10  se  $U \in Visited$  então
11     $tamCycle \leftarrow iterCur - iterLast$ ;
12    Incrementa o número de visitas de  $U$ ;
13    Atualiza a  $iterLast$  de  $U$ ;
14    se  $tamCycle < CYCLE\_MIN$  então
15       $movingAvg \leftarrow 0.9 * movingAvg + 0,1 * tamCycle$ ;
16       $tabuTenure \leftarrow \min(tabuTenure * INCREASE, MAX\_TENURE)$ ;
17       $countLastSizeChange \leftarrow 0$ ;
18    fim
19    se número de visitas a  $U > MAX\_VISITS$  e  $U \notin OftenRepeated$  então
20       $OftenRepeated \leftarrow U$ ;
21       $chaotic ++$ ;
22      se  $chaotic > CHAOS$  então
23         $escape \leftarrow TRUE$ ;
24      fim
25    fim
26  fim
27  senão
28     $Visited \leftarrow U$ ;
29    número de Visitas de  $U \leftarrow 1$ ;
30     $iterLast \leftarrow iterCur$ ;
31    se  $countLastSizeChange > movingAvg$  então
32       $tabuTenure \leftarrow \max(tabuTenure * DECREASE, MIN\_TENURE)$ ;
33       $countLastSizeChange \leftarrow 0$ ;
34    fim
35  fim
36  se  $escape = TRUE$  então
37     $Visited \leftarrow \emptyset$ ;
38     $chaotic \leftarrow 0$ ;
39    Realiza determinado número de movimentos aleatórios em  $U$ .
40  fim
41 fim

```

cardinalidade deste conjunto $OftenRepeated$ e o valor da variável $chaotic$ se tornou maior que a constante $CHAOS$. Se isso acontece, então o mecanismo de escape é disparado para

se mover aleatoriamente da solução atual.

Se a solução atual está sendo visitada pela primeira vez, então o mecanismo de reação atua reduzindo o valor do tabu tenure, no caso em que o valor da variável *movingAvg* é menor que o da variável *countLastSizeChange*. Isto é feito através da multiplicação do tabu tenure por uma constante DECREASE ($0 < \text{DECREASE} < 1$). Além disso, todas as operações com respeito a primeira visita a uma solução são então executadas, de forma que visitas subsequentes seja detectadas. A implementação também usa as constantes MIN_Tenure e MAX_Tenure para limitar o valor do tabu tenure, evitando excessivamente pequenos ou grandes que são indesejados.

Analisando o algoritmo descrito anteriormente, fica claro que o mecanismo de reação de forma estrita possivelmente irá demandar uma grande quantidade de memória e/ou uma estrutura de dados muito poderosa que lide com compressão e tenha um tempo de acesso rápido para manter o conjunto de soluções visitadas. Portanto, ao invés de mantermos um conjunto real de soluções, é mantido uma *hash table* que tem como função de hash uma string parâmetro é obtida pela concatenação só número de vértices cobertos por exatamente i vértices cobertos por exatamente i vértices da solução atual, tendo i variando de 1 a 6.

5 Resultados Computacionais e Conclusão

Os códigos foram implementados em C++ e os experimentos foram executados em um processador *Intel Core i7* 3.3Ghz. Os testes computacionais foram realizados para os códigos $q = 2$. O número de iterações do algoritmo foi de 20.000 iterações para todas as instâncias.

Os parâmetros do algoritmo foram determinados empiricamente, de forma a serem ajustados de com o tamanho das instâncias. Para o fator de penalidade foram adotados os valores 0.1, 1.0 e 0.001 para MIN_ALPHA, MAX_ALPHA e STEP_ALPHA, respectivamente. Os principais parâmetros são: MAX_VISITS, CHAOS, CYCLE_MIN, MIN_TENURE, MAX_TENURE, INCREASE e DECREASE. Sendo os valores 10, 5 e 8 para os três primeiros parâmetros respectivamente. Os valores do tabu tenure variam de 20 a 400. Por fim, os valores dos parâmetros para o aumento ou redução da tabu tenure foram 1.2 e 0.9, respectivamente.

A Tabela 1 apresenta os resultados encontrados pelos algoritmos propostos. A coluna $|V_q^{ms}|$ contém o número de vértices do grafo associado a \mathbb{Z}_q^{ms} . As colunas LB e UB apresentam, respectivamente, os melhores limites inferiores e superiores encontrados utilizando os resultados da Seção 2 e outros resultados apresentados em Castoldi (2012). Os resultados dos algoritmos propostos, Busca Tabu Clássica e Busca Tabu Reativa, são apresentados nas respectivas colunas AUB. Por fim, têm-se as colunas $t_{min}(s)$ e $t_{max}(s)$ referentes ao tempo mínimo e máximo, em segundos, das iterações do algoritmo.

O tempo de execução para ambos algoritmos são próximos, mas a Busca Tabu Clássica é um pouco mais rápida que a Busca Tabu Reativa. A Busca Tabu com o mecanismo de reação apresentou melhor limite superior para 10 instâncias, quando comparada à Busca Tabu Clássica. O código $G(2, 6, 2, 4)$ apresentou maior diferença nos limites, melhorando de 176 para 161 e, no código $G(3, 4, 2, 3)$, de 160 para 128.

Neste trabalho, nenhum limite exato foi encontrado para os problemas de cobertura testados, porém muitos deles estão entre 2 a 3 possibilidades, indicando assim possíveis limites exatos. Os melhores casos são para $G(3, 2, 2, 3)$, $G(4, 2, 2, 4)$ e $G(4, 3, 2, 8)$ nos quais tem-se apenas 2 possibilidades para o limite exato. Para os 11 códigos seguintes, há 3 possibilidades para o limite exato: $G(2, 3, 2, 3)$, $G(2, 4, 2, 5)$, $G(2, 5, 2, 7)$, $G(2, 6, 2, 9)$, $G(3, 3, 2, 3)$, $G(3, 3, 2, 5)$, $G(3, 3, 2, 6)$, $G(3, 2, 4, 8)$, $G(3, 4, 2, 9)$, $G(4, 2, 2, 2)$ e $G(4, 2, 2, 3)$.

Tabela 1: Resultados para a cobertura $K_2^{RT}(m, s, R)$: Grafo $G(m, s, 2, R)$

m	s	R	$ V_q^{ms} $	LB	UB	Busca Tabu Clássica			Busca Tabu Reativa		
						AUB	$t_{min(s)}$	$t_{max(s)}$	AUB	$t_{min(s)}$	$t_{max(s)}$
2	2	1	16	6	8	8	0.03	0.05	8	0.08	0.11
2	3	1	64	22	32	32	0.09	0.10	32	0.17	0.18
2	3	2	64	8	12	12	0.29	0.31	12	0.33	0.36
2	3	3	64	4	8	6	0.48	0.53	6	0.70	0.74
2	4	1	256	86	128	128	0.36	0.40	128	0.50	0.55
2	4	2	256	32	48	48	0.91	0.98	48	1.02	1.12
2	4	3	256	13	32	24	1.92	2.04	24	2.17	2.34
2	4	4	256	6	12	10	2.90	3.02	10	0.33	0.36
2	4	5	256	4	8	6	3.29	3.57	6	0.41	0.46
2	5	1	1024	342	512	512	2.84	3.10	512	2.34	2.60
2	5	2	1024	128	192	192	4.24	4.50	192	4.74	5.10
2	5	3	1024	52	128	96	7.49	8.21	96	9.12	9.98
2	5	4	1024	22	48	43	10.41	10.61	40	12.14	12.58
2	5	5	1024	10	32	21	17.27	17.61	20	19.04	19.74
2	5	6	1024	6	12	11	25.29	25.80	11	30.60	33.88
2	5	7	1024	4	8	6	38.70	39.28	6	46.41	49.82
2	6	1	4096	1366	2048	2048	16.27	16.77	2048	18.37	19.29
2	6	2	4096	512	768	768	18.92	19.59	768	22.86	24.21
2	6	3	4096	205	512	384	33.82	34.96	384	40.12	41.22
2	6	4	4096	86	192	176	45.04	46.44	161	5.50	5.72
2	6	5	4096	37	128	88	72.88	76.37	83	8.76	9.14
2	6	6	4096	16	48	40	137.95	139.03	40	16.51	17.16
2	6	7	4096	10	32	22	260.38	261.72	22	31.26	32.65
2	6	8	4096	6	12	11	48.45	52.18	11	54.98	56.55
2	6	9	4096	4	8	6	77.24	81.37	6	86.75	89.03
3	2	2	64	5	12	8	0.03	0.05	8	0.53	0.58
3	2	3	64	3	8	4	0.05	0.06	4	0.72	0.79
3	3	2	512	40	96	64	0.26	0.29	64	3.17	3.30
3	3	3	512	14	65	16	0.46	0.49	16	5.48	5.73
3	3	4	512	7	16	12	0.73	0.79	12	0.90	0.94
3	3	5	512	4	12	6	1.13	1.22	6	1.38	1.45
3	3	6	512	2	8	4	1.71	1.83	4	2.08	2.32
3	4	2	4096	316	768	512	2.77	2.92	512	34.06	34.44
3	4	3	4096	108	512	160	4.41	4.73	128	51.14	53.70
3	4	4	4096	40	128	80	7.38	7.84	80	8.82	8.99
3	4	5	4096	19	96	41	13.00	14.13	41	15.44	16.17
3	4	6	4096	10	48	24	28.11	29.69	23	32.05	34.78
3	4	7	4096	5	16	12	52.41	54.41	12	60.14	61.48
3	4	8	4096	3	12	8	84.31	87.33	6	95.55	98.90
3	4	9	4096	2	8	4	115.51	120.78	4	132.97	137.58
4	2	1	256	52	64	64	0.06	0.07	64	0.97	1.02
4	2	2	256	14	48	16	0.16	0.18	16	2.32	2.40
4	2	3	256	6	16	8	0.24	0.27	8	3.58	3.71
4	2	4	256	3	8	4	0.40	0.43	4	0.55	0.58
4	3	1	4096	820	1024	1024	1.74	1.89	1024	19.58	20.60
4	3	2	4096	216	768	256	3.41	3.62	256	35.61	38.91
4	3	3	4096	66	256	129	5.41	5.86	124	65.08	66.87
4	3	4	4096	26	128	56	9.67	10.32	56	11.54	12.01
4	3	5	4096	12	48	26	20.85	22.11	26	24.39	25.43
4	3	6	4096	6	32	15	44.41	46.43	14	52.10	54.08
4	3	7	4096	4	16	8	71.68	74.14	8	84.54	85.36
4	3	8	4096	3	12	5	109.43	114.76	4	125.63	126.28

Melhorar o limite superior para R -coberturas em espaços de Hamming e RT-espaços é frequentemente uma tarefa difícil. Uma evidência disso é que para várias instâncias $K_q(n, R)$ o valor exato ainda não é conhecido. Assim, os resultados obtidos evidenciam a importância de uma abordagem computacional como feita neste trabalho, visto que o algoritmo foi capaz de melhorar o limite superior para 78% dos 52 códigos testados.

6 Agradecimentos

Ao CNPq pelo financiamento parcial dos autores Mayra C. Albuquerque e Marcus Poggi. Processo 141225/2012-2. E à CAPES pelo apoio financeiro aos autores André G. Castoldi e Emerson L. do Monte Carmelo.

Referências

- Bastos, M. e Ribeiro, C. (2002). Reactive tabu search with path-relinking for the steiner problem in graphs. In *Essays and Surveys in Metaheuristics*, volume 15 of *Operations Research/Computer Science Interfaces Series*, pages 39 – 58. Springer US.
- Battiti, R. e Tecchioli, G. (1994). The reactive tabu search. *ORSA Journal on Computing*, 6(2):126–40.
- Carnielli, W. A. (1985). On covering and coloring problems for rook domains. *Discrete Mathematics*, 57(1-2):9 – 16.
- Carnielli, W. A., Carmelo, E. L. M., e Poggi, M. (1995). Upper bounds for minimum covering codes by tabu search. *II Oficina Nacional de Problemas Combinatórios: Teoria, Algoritmos e Aplicações*, pages 50–58.
- Castoldi, A. e Carmelo, E. M. (2013). Coverings in rosenbloom-tsfasman spaces.
- Castoldi, A. G. (2012). Coberturas em espaços de rosenbloom-tsfasman. Master's thesis, Universidade Estadual de Maringá.
- Dougherty, S. T. e Skrikanov, M. M. (2002a). Macwilliams duality and the rosenbloom-tsfasman metric. *Moscow Mathematical Journal*, 2:83 – 89.
- Dougherty, S. T. e Skrikanov, M. M. (2002b). Maximum distance separable codes in the ρ metric over arbitrary alphabets. *J. Algebraic Comb.*, 16(1):71–81.
- Garey, M. R. e Johnson, D. S. (1979). *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman & Co., New York, NY, USA.
- Glover, F. e Laguna, M. (1997). *Tabu Search*. Kluwer Academic Publishers, Norwell, MA, USA.
- Mendes, C., Monte Carmelo, E. L., e Poggi, M. (2010). Bounds for short covering codes and reactive tabu search. *Discrete Appl. Math.*, 158(5):522–533.
- Nanry, W. P. e Barnes, J. W. (2000). Solving the pickup and delivery problem with time windows using reactive tabu search. *Transportation Research Part B: Methodological*, 34(2):107–121.
- Osman, I. H. e Wassan, N. A. (2002). A reactive tabu search meta-heuristic for the vehicle routing problem with back-hauls. *Journal of Scheduling*, 5(4):263–285.
- Ozen, M. e Siap, I. (2004). The complete weight enumerator for codes over $m_{n \times s}(r)$. *Applied Mathematics Letters*, 17:65–69.

- Ozen, M. e Siap, I. (2006). Linear codes over $\mathbb{F}_q[u]/(u^s)$ with respect to the rosenbloom-tsfasman metric. *Designs, Codes and Cryptography*, 38:17–29.
- Quistorff, J. (2007). On rosenbloom and tsfasman’s generalization of the hamming space. *Discrete Mathematics*, 307:2514–2524.
- Rosenbloom, M. Y. e Tsfasman, M. A. (1997). Codes for the m-metric. *Problemy Peredachi Informatsii*, 33(1):55–63.
- Taussky, O. e Todd, J. (1948). Covering theorems for group. *Annales Polonici Mathematici*, 21:303–305.
- Yildiz, B., Siap, I., Bilgin, T., e Yesilot, G. (2010). The covering problem for finite rings with respect to the rt-metric. *Applied Mathematics Letters*, 23(9):988 – 992.