t-Linearizalização de Funções Quadráticas de Variáveis Binárias

Pablo Soares
Universidade Federal do Ceará
Campus de Russas, CEP 62900-000, Russas - CE, Brasil.
pablo.soares@ufc.br

Manoel Campêlo, Carlos Diego Rodrigues
Universidade Federal do Ceará, Departamento de Estatística e Matática Aplicada
Campus do Pici, Bloco 910, CEP 60440-900, Fortaleza - CE, Brasil.
{mcampelo,diego}@lia.ufc.br

Philippe Michelon
Université d'Avignon et des Pays de Vaucluse
74 rue Louis Pasteur, 84029, Avignon cedex 1, France.
philippe.michelon@univ-avignon.fr

RESUMO

Generalizamos a técnica da t-linearização para problemas quadráticos 0-1 com coeficientes quadráticos arbitrários. O problema linearizado equivalente inclui uma única variável extra, porém adiciona um número exponencial de restrições. Mostramos que tais restrições podem ser separadas em tempo polinomial. Aplicamos a t-linearização ao modelo quadrático 0-1 clássico para o problema do corte máximo em um grafo. Comparamos computacionalmente o modelo obtido com a linearização clássica e a formulação de programação semidefinda do problema. Experimentos computacionais mostram que a t-linearização é competitiva com a linearização clássica em termos de tempo de processamento e *gaps* de integralidade, sendo estes últimos sensíveis ao tamanho e à desidade do grafo de entrada. Por outro lado, a programação semidefinida gera limites bastante superiores a um custo computacional bem mais elevado.

PALAVRAS CHAVE. t-Linearização, Programação Quadrática 0-1, Corte Máximo.

Otimização Combinatória, Programação Matemática

ABSTRACT

We generalize the t-linearization technique for 0-1 quadratic problems with arbitrary quadratic coefficients. The linearized problem adds a unique extra variable and an exponencial number of constraints. We show that such constraints can be separated in polynomial time. We apply the t-linearization to the classical 0-1 quadratic model for the max-cut problem. We computationally compare the obtained model with the classical linearization and the semidefinite programming model. Computational experiments show that the t-linearization is competitive with the standard linearization in terms of processing time and integrality gaps, the latter being sensible to the input graph size and density. On the other hand, the SDP provide much better bounds on much higher computational costs.

KEYWORDS. t-Linearization, 0-1 Quadratic Programming, Max Cut.

Combinatorial Optimization, Mathematical Programming

Introdução

A não-linearidade em problemas de programação inteira é normalmente tratada com o uso de técnicas envolvendo uma aproximação linear por partes [Dantzig, 1963; Hu, 1969] ou a transformação da função não linear em uma função polinomial, a seguir convertida em uma função linear de variáveis 0-1 [Balas, 1964; Hammer e Rudeanu, 1968]. Na maioria das vezes, a não-linearidade em problemas de programação inteira aparece já na forma polinomial, sendo que um número significativo dos casos envolve apenas termos de segunda ordem [Glover, 1975].

Neste trabalho, concentramo-nos em problemas quadráticos 0-1, que estão postos entre os problemas NP-Difíceis mais desafiadores em otimização combinatória [Gary e Johnson, 1979]. O problema quadrático (PQ), base para o estudo de outros problemas dessa classe, pode ser escrito da seguinte forma:

$$(PQ) \quad \max \left\{ \sum_{i=1}^{n} \sum_{j=i+1}^{n} c_{ij} x_i x_j : x \in \{0,1\}^n \right\},$$

onde c_{ij}, $1 \leq i < j \leq n$, são coeficientes reais arbitrários. Por conveniência, definimos os coeficientes adicionais $c_{ij} = c_{ji}$ e $c_{ii} = 0$, $1 \leq j \leq i \leq n$.

Devido à ampla gama de problemas quadráticos da literatura e à dificuldade em resolvê-los de forma exata, faz-se necessário a obtenção de limites (inferior e superior) que possam auxiliar o desenvolvimento de métodos exatos. Gueye e Michelon [2009] dividiram esses limites em 04 grupos, de acordo com as técnicas usadas para obtê-los: Programação Semidefinida [Goemans e Williamson, 1995; Rendl et al., 2010], Decomposição Lagrageana [Chardaire e Sutter, 1995; Elloumi et al., 2000], métodos posiform [Boros e Hammer, 2002; Billionnet e Sutter, 1994] e técnicas de linearização [Fortet, 1960; Rodrigues, 2010].

Dentre as técnicas de linearização, que consistem em transformar a função quadrática em uma função linear, Rodrigues et al. [2012] destacam duas abordagens: (i) linearização com acréscimo de um número polinomial de variáveis e restrições lineares; (ii) linearização sem adição de variáveis, mas com um número exponencial de restrições lineares. No primeiro grupo, encontra-se o procedimento clássico de linearização [Fortet, 1960], que consiste em substituir cada produto $x_i x_j$ por uma variável não negativa z_{ij} e adicionar restrições $z_{ij} \leq x_i, \quad z_{ij} \leq x_j, \quad z_{ij} \geq x_i + x_j - 1$. Com isso, podemos reescrever (PQ) da seguinte forma equivalente:

$$(PQ)_{LC} \quad \max \left\{ \sum_{i=1}^{n} \sum_{j=i+1}^{n} c_{ij} z_{ij} : z_{ij} \leq x_i, z_{ij} \leq x_j, z_{ij} \geq x_i + x_j - 1, x \in \{0,1\}^n \right\}.$$

No segundo grupo, está a t-linearização, proposta no contexto do Problema Quadrático da Mochila (PQM) [Rodrigues, 2010; Rodrigues et al., 2012]. Essa abordagem consiste basicamente de duas etapas. Primeiro, substitui-se o termo quadrático da função objetivo de (PQ) por uma variável real t, que é limitada superiormente pela expressão quadrática, com a inclusão de uma restrição adicional. Depois, essa restrição quadrática é substituída por um conjunto exponencial de restrições lineares, equivalentes quando os coeficientes do termo quadrático são não-negativos. Essa propriedade, válida em (PQM), é fortemente usada em [Rodrigues et al., 2012].

Na próxima seção, revisitamos a t-linearização para termos quadráticos com coeficientes não-negativos, apresentando outras demonstrações para resultados de [Rodrigues et al., 2012]. Na Seção 2, estendemos tais resultados para coeficientes arbitrários. Basicamente, eles se referem à descrição das restrições lineares que substituem a quadrática, bem como a uma forma eficiente de separá-las. Já na Seção 4, aplicamos tal desenvolvimento ao problema do corte máximo (Max-Cut), modelado na forma (PQ). Experimentos computacionais para Max-Cut, comparando a t-linearização, a linerarização clássica e programação semidefinida, são apresentados na Seção 5. Conclusões resumidas fecham o artigo.

t–linearização para coeficientes não negativos

Ao longo desta seção, consideramos $c_{ij} \geq 0$, $1 \leq i < j \leq n$. Com o auxílio de uma variável adicional t, podemos reescrever (PQ) como

$$(PQ)_t \quad \max\{t : (x,t) \in Z\}, \quad \text{onde} \quad Z = \left\{(x,t) \in \mathbb{B}^n \times \mathbb{R} : t \leq \sum_{i=1}^{n}\sum_{j=1}^{i-1} c_{ji}x_i x_j\right\}.$$

Seja S_n o conjunto de todas as permutações de $\{1, \ldots, n\}$. O conjunto Z pode ser expresso por restrições lineares, como segue:

Teorema 1. $Z = Z' := \left\{(x,t) \in \mathbb{B}^n \times \mathbb{R} : t \leq \sum_i \sum_{j<i} c_{\pi(j)\pi(i)} x_{\pi(i)} \ \forall \pi \in S_n\right\}.$

Demonstração. Primeiro, note que

$$\sum_{i=1}^{n}\sum_{j=1}^{i-1} c_{ji}x_i x_j = \sum_{i=1}^{n}\sum_{j=i}^{i-1} c_{\pi(j)\pi(i)} x_{\pi(i)} x_{\pi(j)} \ \forall \pi \in S_n. \tag{1}$$

Se $(x,t) \in Z$, então temos que $(x,t) \in Z'$ devido a (1) e

$$\sum_i \sum_{j<i} c_{\pi(j)\pi(i)} x_{\pi(i)} x_{\pi(j)} \leq \sum_i \sum_{j<i} c_{\pi(j)\pi(i)} x_{\pi(i)} \ \forall \pi \in S_n.$$

Suponha agora $(x,t) \in Z'$ e seja $\pi \in S_n$ que ordena as componentes de x em ordem não-crescente, ou seja, $x_{\pi(i)} \geq x_{\pi(j)}$ se $i \leq j$. Observe que $(x_{\pi(1)}, \ldots, x_{\pi(n)}) = (1, \ldots, 1, 0, \ldots, 0)$. Como $(x,t) \in Z$, temos que

$$t \leq \sum_i \sum_{j<i} c_{\pi(j)\pi(i)} x_{\pi(i)} = \sum_i \sum_{j<i} c_{\pi(j)\pi(i)} x_{\pi(i)} x_{\pi(j)},$$

onde a igualdade decorre da ordenação das entradas de x por π. $\square$

Pelo Teorema (1), podemos substituir Z por Z' na descrição de $(PQ)_t$. Na verdade, Rodrigues [2010] mostra que a integralidade das variáveis x pode ser descartada na definição de Z', pois sua envoltória convexa é o poliedro relaxado. Mais ainda, a separação das restrições lineares que definem Z' pode ser feita em tempo polinomial, como mostraremos a seguir.

Lema 1. *Seja $\hat{x} \in [0,1]^n$. A solução do problema*

$$\min_{\pi \in S_n} \sum_i \sum_{j<i} c_{\pi(j)\pi(i)} \hat{x}_{\pi(i)} \tag{2}$$

ocorre na permutação $\hat{\pi} \in S_n$ que ordena as componentes de $\hat{x}$ em ordem não-crescente, ou seja, tal que

$$\hat{x}_{\hat{\pi}(i)} \geq \hat{x}_{\hat{\pi}(j)} \ \forall 1 \leq i \leq j \leq n. \tag{3}$$

Demonstração. Seja $\pi \in S_n$ uma solução ótima para (2). Suponha que π não satisfaz a propriedade (3). Seja, então, $l \geq i$ o maior inteiro tal que as primeiras l posições de $\hat{x}$ segundo π estão ordenadas, ou seja, $\hat{x}_{\pi(1)} \geq \hat{x}_{\pi(2)} \geq \ldots \geq \hat{x}_{\pi(l)} < \hat{x}_{\pi(l+1)}$. Seja $k \leq l$ tal que

$$\hat{x}_{\pi(1)} \geq \ldots \geq \hat{x}_{\pi(k-1)} \geq \hat{x}_{\pi(l+1)} > \hat{x}_{\pi(k)} \geq \hat{x}_{\pi(l)}. \tag{4}$$

Defina uma nova permutação π' que ordena as entradas de $\hat{x}$ como em (4), ou seja, tal que

$$\pi'(i) = \begin{cases} \pi(i), & i = 1\dots k-1,\ i = l+2\dots n, \\ \pi(l+1), & i = k, \\ \pi(i-1), & i = k+1\dots l+1. \end{cases}$$

Então

$$\begin{aligned} \sum_i \sum_{j<i} c_{\pi'(j)\pi'(i)}\hat{x}_{\pi'(i)} &= \sum_i \sum_{j<i} c_{\pi(j)\pi(i)}\hat{x}_{\pi(i)} + \sum_{i=k}^{l} c_{\pi(l+1)\pi(i)}\hat{x}_{\pi(i)} - \sum_{i=k}^{l} c_{\pi(i)\pi(l+1)}\hat{x}_{\pi(l+1)} \\ &= \sum_i \sum_{j<i} c_{\pi(j)\pi(i)}\hat{x}_{\pi(i)} + \sum_{i=k}^{l} \underbrace{c_{\pi(i)\pi(l+1)}}_{\geq 0} \underbrace{\left(\hat{x}_{\pi(i)} - \hat{x}_{\pi(l+1)}\right)}_{<0} \\ &\leq \sum_i \sum_{j<i} c_{\pi(j)\pi(i)}\hat{x}_{\pi(i)}. \end{aligned}$$

Logo, π' também é ótimo para (2) e satisfaz a propriedade (3) até pelo menos $l+1$, ou seja, $\hat{x}_{\pi'(1)} \geq \hat{x}_{\pi'(2)} \geq \ldots \geq \hat{x}_{\pi'(l+1)}$. Repetindo o processo a partir de π', em um número fixo de passos obtemos uma permutação $\hat{\pi}$ satisfazendo (3) e que também é ótima. $\square$

Extensão da t-Linearização para Coeficientes Arbitrários

Nesta seção, consideramos coeficientes arbitrários $c_{ij} = c_{ji}$, $1 \leq i, j \leq n$. Por conveniência, vamos multiplicar a função objetivo de (PQ) por 2, obtendo

$$(PQ) \quad \max\{F(x) : x \in \mathbb{B}^n\} \quad \text{onde} \quad F(x) = \sum_{i=1}^{n} \sum_{j=i+1}^{n} 2c_{ij}x_i x_j = \sum_{i=1}^{n} \sum_{j=1}^{i-1} 2c_{ji}x_i x_j.$$

Denote $a^+ = \max\{0, a\}$, $a^- = \min\{0, a\}$, $\forall a \in \mathbb{R}$, e $\overline{a} = 1 - a$, $\forall a \in [0, 1]$. Note que

$$\begin{aligned} F(x) &= \sum_{i=1}^{n} \sum_{j=1}^{i-1} 2c_{ji}^{+}x_i x_j + \sum_{i=1}^{n} \sum_{j=1}^{i-1} \underbrace{2c_{ji}^{-}x_i x_j}_{c_{ji}^{-}x_i(1-\overline{x}_j)+c_{ji}^{-}(1-\overline{x}_i)x_j} \\ &= Q(x) + L(x), \end{aligned}$$

onde

$$\begin{aligned} Q(x) &= \sum_{i=1}^{n} \sum_{j=1}^{i-1} 2c_{ji}^{+}x_i x_j + \sum_{i=1}^{n} \sum_{j=1}^{i-1} -c_{ji}^{-}(x_i\overline{x}_j + \overline{x}_i x_j), \\ L(x) &= \sum_{i=1}^{n} \sum_{j=1}^{i-1} c_{ji}^{-}(x_i + x_j) = \sum_i \sum_{j<i} c_{ji}^{-}x_i + \sum_i \sum_{j>i} c_{ji}^{-}x_i = \sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij}^{-}x_i. \end{aligned}$$

Assim, podemos reescrever (PQ) como

$$(PQ)_t \quad \max\{t + L(x) : (x, t) \in Z\} \quad \text{onde} \quad Z = \{(x, t) \in \mathbb{B}^n \times \mathbb{R} : t \leq Q(x)\}.$$

A restrição quadrática que define Z pode ser substituída por expressões lineares, como mostraremos a seguir. Denote por $\mathcal{N} = \{(N_0, N_1) : N_0 \cup N_1 = N, N_0 \cap N_1 = \emptyset\}$ o conjunto de partições de $N = \{1, \ldots, n\}$.

Teorema 2. *O conjunto Z é igual a*

$$Z' := \left\{ (x,t) \in \mathbb{B}^n \times \mathbb{R} : t \leq \sum_{i\in N} \sum_{\substack{\pi(j)\in N_1 \\ j<i}} 2c^+_{\pi(j)\pi(i)} x_{\pi(i)} + \sum_{i\in N} \sum_{\substack{\pi(j)\in N_0 \\ j<i}} 2c^+_{\pi(j)\pi(i)} x_{\pi(j)} + \right.$$
$$\left. \sum_{i\in N} \sum_{\substack{\pi(j)\in N_1 \\ j<i}} -c^-_{\pi(j)\pi(i)}(\overline{x}_{\pi(i)} + \overline{x}_{\pi(j)}) + \sum_{i\in N} \sum_{\substack{\pi(j)\in N_0 \\ j<i}} -c^-_{\pi(j)\pi(i)}(x_{\pi(i)} + x_{\pi(j)}) \ \forall \pi \in S_n \ \forall (N_0, N_1) \in \mathcal{N} \right\}.$$

Demonstração. Seja $(x,t) \in Z$ e defina $\overline{x} = \mathbf{1} - x$. Então

$$t \leq Q(x) = \underbrace{\sum_{i\in N} \sum_{j<i} 2c^+_{ji} x_i x_j}_{A} + \underbrace{\sum_{i\in N} \sum_{j<i} (-c^-_{ji})[x_i\overline{x}_j + \overline{x}_i x_j]}_{B}.$$

Observe que A soma $2c^+_{ji}x_ix_j$ para todos os pares não ordenados ij com $c_{ij} > 0$, e B soma $-c^-_{ji}(x_i\overline{x}_j + \overline{x}_i x_j)$ para todos os pares não ordenados ij com $c_{ij} < 0$. Para $\pi \in S_n$, temos

$$Q(x) = \underbrace{\sum_{i\in N} \sum_{j<i} 2c^+_{\pi(j)\pi(i)} x_{\pi(i)} x_{\pi(j)}}_{C} + \underbrace{\sum_{i\in N} \sum_{j<i} (-c^-_{\pi(j)\pi(i)})[x_{\pi(i)}\overline{x}_{\pi(j)} + \overline{x}_{\pi(i)} x_{\pi(j)}]}_{D}.$$

Para $(N_0, N_1) \in \mathcal{N}$ partição de N, temos

$$\begin{aligned}
C &= \sum_{i\in N} \sum_{\substack{j<i \\ \pi(j)\in N_1}} \underbrace{(2c^+_{\pi(j)\pi(i)} x_{\pi(i)})}_{\geq 0} \underbrace{x_{\pi(j)}}_{\leq 1} + \sum_{i\in N} \sum_{\substack{j<i \\ \pi(j)\in N_0}} \underbrace{(2c^+_{\pi(j)\pi(i)} x_{\pi(j)})}_{\geq 0} \underbrace{x_{\pi(i)}}_{\leq 1} \\
&\leq \sum_{i\in N} \sum_{\substack{j<i \\ \pi(j)\in N_1}} 2c^+_{\pi(j)\pi(i)} x_{\pi(i)} + \sum_{i\in N} \sum_{\substack{j<i \\ \pi(j)\in N_0}} 2c^+_{\pi(j)\pi(i)} x_{\pi(j)} \\
D &= \sum_{i\in N} \sum_{\substack{j<i \\ \pi(j)\in N_1}} \underbrace{-c^-_{\pi(j)\pi(i)}}_{\geq 0} \underbrace{[x_{\pi(i)}\overline{x}_{\pi(j)} + \overline{x}_{\pi(i)} x_{\pi(j)}]}_{\leq \overline{x}_{\pi(i)} + \overline{x}_{\pi(j)}} + \sum_{i\in N} \sum_{\substack{j<i \\ \pi(j)\in N_0}} \underbrace{-c^-_{\pi(j)\pi(i)}}_{\geq 0} \underbrace{[x_{\pi(i)}\overline{x}_{\pi(j)} + \overline{x}_{\pi(i)} x_{\pi(j)}]}_{\leq x_{\pi(i)} + x_{\pi(j)}} \\
&\leq \sum_{i\in N} \sum_{\substack{j<i \\ \pi(j)\in N_1}} -c^-_{\pi(j)\pi(i)}(\overline{x}_{\pi(i)} + \overline{x}_{\pi(j)}) + \sum_{i\in N} \sum_{\substack{j<i \\ \pi(j)\in N_0}} -c^-_{\pi(j)\pi(i)}(x_{\pi(i)} + x_{\pi(j)})
\end{aligned}$$

Como $t \leq Q(x) = C + D$ e $\pi \in S_n$, $(N_0, N_1) \in \mathcal{N}$ são arbitrírios, concluímos que $(x,t) \in Z'$.

Suponha agora $(x,t) \in Z'$. Seja $\pi \in S_n$. Defina $N_0 = \{j \in N : x_j = 0\}$ e $N_1 = \{j \in N : x_j = 1\}$. Temos que

$$\begin{aligned}
Q(x) &= \sum_{i\in N} \sum_{j<i} 2c^+_{\pi(j)\pi(i)} x_{\pi(i)} x_{\pi(j)} + \sum_{i\in N} \sum_{j<i} (-c^-_{\pi(j)\pi(i)})[x_{\pi(i)}\overline{x}_{\pi(j)} + \overline{x}_{\pi(i)} x_{\pi(j)}] \\
&= \sum_{i\in N} \sum_{\substack{j<i \\ \pi(j)\in N_1}} 2c^+_{\pi(j)\pi(i)} x_{\pi(i)} + \sum_{i\in N} \sum_{\substack{j<i \\ \pi(j)\in N_1}} -c^-_{\pi(j)\pi(i)}(1 - x_{\pi(i)}) + \sum_{i\in N} \sum_{\substack{j<i \\ \pi(j)\in N_0}} -c^-_{\pi(j)\pi(i)} x_{\pi(i)}
\end{aligned}$$

Por outro lado, como $(x,t) \in Z'$, então

$$t \leq \sum_{i\in N} \sum_{\substack{\pi(j)\in N_1 \\ j<i}} -c^-_{\pi(j)\pi(i)}(1 - x_{\pi(i)}) + \sum_{i\in N} \sum_{\substack{\pi(j)\in N_0 \\ j<i}} -c^-_{\pi(j)\pi(i)} x_{\pi(i)} + \sum_{i\in N} \sum_{\substack{\pi(j)\in N_1 \\ j<i}} 2c^+_{\pi(j)\pi(i)} x_{\pi(i)} = Q(x).$$

Logo, $(x,t) \in Z$. $\square$

O Teorema 2 generaliza, para coeficientes arbitrários, o resultado de Rodrigues et al. [2012]. Como consequência, podemos transformar o problema quadrático $(PQ)_t$ e, portanto (PQ), em um problema de programação linear inteira equivalente, com a adição de uma variável extra e um número exponencial de $2^n \cdot n!$ restrições. Adicionalmente, mostraremos a seguir, que a separação dessas desigualdades pode ser feita em tempo polinomial. Para isso, para $\pi \in S_n$, $(N_0, N_1) \in \mathcal{N}$ e $x \in [0,1]^n$, defina

$$\begin{aligned}\Theta(\pi, N_0, N_1; x) &= \sum_{i \in N} \sum_{\substack{\pi(j) \in N_1 \\ j<i}} \alpha_{\pi(j)\pi(i)}(\overline{x}_{\pi(i)} + \overline{x}_{\pi(j)}) + \sum_{i \in N} \sum_{\substack{\pi(j) \in N_0 \\ j<i}} \alpha_{\pi(j)\pi(i)}(x_{\pi(i)} + x_{\pi(j)}) \\ &+ \sum_{i \in N} \sum_{\substack{\pi(j) \in N_1 \\ j<i}} \beta_{\pi(j)\pi(i)} x_{\pi(i)} + \sum_{i \in N} \sum_{\substack{\pi(j) \in N_0 \\ j<i}} \beta_{\pi(j)\pi(i)} x_{\pi(j)},\end{aligned}$$

onde $\alpha_{\pi(j)\pi(i)} = -c^-_{\pi(j)\pi(i)} \geq 0$ e $\beta_{\pi(j)\pi(i)} = 2c^+_{\pi(j)\pi(i)} \geq 0\ \forall i \in N\ \forall j \in N\ j < i$.

O problema de separação consiste em, dado $x^* \in [0,1]^n$, resolver:

$$\min_{(N_0,N_1) \in \mathcal{N}} \min_{\pi \in S_n} \Theta(\pi, N_0, N_1; x^*) \tag{5}$$

Lema 2. *Para $x^* \in [0,1]^n$ e $(\hat{N}_0, \hat{N}_1) \in \mathcal{N}$, a permutação $\hat{\pi} \in S_n$ que minimiza $\Theta(\pi, \hat{N}_0, \hat{N}_0; x^*)$ é tal que*

$$\hat{y}_{\hat{\pi}(1)} \geq \hat{y}_{\hat{\pi}(2)} \geq \ldots \geq \hat{y}_{\hat{\pi}(n)} \tag{6}$$

onde $\hat{y}_j = \begin{cases} x^*_j, & j \in N_1, \\ \overline{x^*_j}, & j \in N_0. \end{cases}$

Demonstração. Fixe $(\hat{N}_0, \hat{N}_1) \in \mathcal{N}$ e $x^* \in [0,1]^n$. Seja $\pi \in S_n$ que minimiza $\Theta(\pi, \hat{N}_0, \hat{N}_0; x^*)$. Suponha que π não satisfaça (6). Seja, então, $l \geq 1$ o máximo inteiro tal que as primeiras l posições de $\hat{y}$ estão ordenadas, segundo π. Então, $\hat{y}_{\pi(1)} \geq \hat{y}_{\pi(2)} \geq \ldots \geq \hat{y}_{\pi(l)} < \hat{y}_{\pi(l+1)}$. Seja $k \leq l$ tal que $\hat{y}_{\pi(1)} \geq \ldots \geq \hat{y}_{\pi(k-1)} \geq \hat{y}_{\pi(l+1)} > \hat{y}_{\pi(k)} \geq \ldots \geq \hat{y}_{\pi(l)}$. Defina uma nova permutação π' que ordena as primeiras $l+1$ entradas de $\hat{y}$, ou seja,

$$\pi'(i) = \begin{cases} \pi(i), & i = 1 \ldots k-1, \text{ ou } i = l+2 \ldots n, \\ \pi(l+1), & i = k, \\ \pi(i-1), & i = k+1 \ldots l+1. \end{cases}$$

Seja $\Delta = \Theta(\pi', \hat{N}_0, \hat{N}_1; x^*) - \Theta(\pi, \hat{N}_0, \hat{N}_1; x^*)$. Primeiro suponha que $\pi(l+1) \in N_1$. Então, $\Delta = B + C$, onde

$$\begin{aligned} B = &\sum_{i=k}^{l} \alpha_{\pi(l+1)\pi(i)}(\overline{x^*}_{\pi(i)} + \overline{x^*}_{\pi(l+1)}) - \sum_{\substack{j=k \\ \pi(j) \in N_1}}^{l} \alpha_{\pi(i+1)\pi(j)}(\overline{x^*}_{\pi(j)} + \overline{x^*}_{\pi(l+1)}) \\ &- \sum_{\substack{j=k \\ \pi(j) \in N_0}}^{l} \alpha_{\pi(l+1)\pi(j)}(x^*_{\pi(j)} + x^*_{\pi(l+1)}), \\ C = &\sum_{i=k}^{l} \beta_{\pi(l+1)\pi(i)} x^*_{\pi(i)} - \sum_{\substack{j=k \\ \pi(j) \in N_1}}^{l} \beta_{\pi(j)\pi(l+1)} x^*_{\pi(l+1)} - \sum_{\substack{j=k \\ \pi(j) \in N_0}}^{l} \beta_{\pi(j)\pi(l+1)} x^*_{\pi(j)}. \end{aligned}$$

Logo,

$$
\begin{aligned}
\Delta = & \sum_{\substack{j=k \\ \pi(j)\in N_0}}^{l} \alpha_{\pi(l+1)\pi(i)}(\overline{x^*}_{\pi(l+1)} + \overline{x^*}_{\pi(j)} - x^*_{\pi(j)} - x^*_{\pi(l+1)}) + \sum_{\substack{j=k \\ \pi(j)\in N_1}}^{l} \beta_{\pi(l+1)\pi(i)}(x^*_{\pi(j)} - x^*_{\pi(l+1)}) \\
= & \sum_{\substack{j=k \\ \pi(j)\in N_0}}^{l} \alpha_{\pi(l+1)\pi(i)} 2(\overline{x^*}_{\underbrace{\pi(j)}_{\in N_0}} - x^*_{\underbrace{\pi(l+1)}_{\in N_1}}) + \sum_{\substack{j=k \\ \pi(j)\in N_1}}^{l} \beta_{\pi(l+1)\pi(i)}(x^*_{\underbrace{\pi(j)}_{\in N_1}} - x^*_{\underbrace{\pi(l+1)}_{\in N_1}}) \\
= & \sum_{\substack{j=k \\ \pi(j)\in N_0}}^{l} \underbrace{\alpha_{\pi(l+1)\pi(i)} 2}_{\geq 0}\underbrace{(\hat{y}_{\pi(j)} - \hat{y}_{\pi(l+1)})}_{<0} + \sum_{\substack{j=k \\ \pi(j)\in N_1}}^{l} \underbrace{\beta_{\pi(l+1)\pi(i)}}_{\geq 0}\underbrace{(\hat{y}_{\pi(j)} - \hat{y}_{\pi(l+1)})}_{<0},
\end{aligned}
$$

ou seja, $\Theta(\pi', \hat{N}_0, \hat{N}_1; x^*) \leq \Theta(\pi, \hat{N}_0, \hat{N}_1; x^*)$. Então π' também minimiza $\Theta(\pi, \hat{N}_0, \hat{N}_1; \hat{x})$ e ordena pelo menos mais uma entrada em $\hat{y}$. Repetindo o processo um número finito de passos, chegamos a uma permutação π que satisfaz (6).

Agora suponha que $\pi(l+1) \in N_0$. Neste caso, $\Delta = E + F$, onde

$$
\begin{aligned}
E = & -\sum_{\substack{j=k \\ \pi(j\in N_1)}}^{l} \alpha_{\pi(l+1)\pi(j)}(\overline{x^*}_{\pi(l+1)} + \overline{x^*}_{\pi(j)}) + \sum_{i=k}^{l} \alpha_{\pi(l+1)\pi(i)}(x^*_{\pi(i)} + x^*_{\pi(l+1)}) \\
& - \sum_{\substack{j=1 \\ \pi(j)\in N_0}}^{l} \alpha_{\pi(l+1)\pi(j)}(x^*_{\pi(l+1)} + x^*_{\pi(j)}), \\
F = & -\sum_{\substack{i=k \\ \pi(j)\in N_1}}^{l} \beta_{\pi(j)\pi(l+1)} x^*_{\pi(l+1)} + \sum_{i=k}^{l} \beta_{\pi(l+1)\pi(i)} x^*_{\pi(l+1)} - \sum_{\substack{j=k \\ \pi(j)\in N_0}}^{l} \beta_{\pi(l+1)\pi(j)} x^*_{\pi(j)}.
\end{aligned}
$$

Portanto,

$$
\begin{aligned}
\Delta = & \sum_{\substack{j=k \\ \pi(j)\in N_1}}^{l} \alpha_{\pi(l+1)\pi(j)}(x^*_{\pi(l+1)} + x^*_{\underbrace{\pi(j)}_{\in N_1}} - \overline{x^*}_{\underbrace{\pi(l+1)}_{\in N_0}} - \overline{x^*}_{\pi(j)}) \\
& + \sum_{\substack{j=k \\ \pi(j)\in N_0}}^{l} \beta_{\pi(j)\pi(l+1)} \overbrace{(x^*_{\underbrace{\pi(l+1)}_{\in N_0}} - x^*_{\underbrace{\pi(j)}_{\in N_0}})}^{\overline{x^*}_{(j)} - \overline{x^*}_{(l+1)}} \\
= & \sum_{\substack{j=k \\ \pi(j)\in N_1}}^{l} \underbrace{\alpha_{\pi(l+1)\pi(j)} 2}_{\geq 0}\underbrace{(\hat{y}_{\pi(j)} - \hat{y}_{\pi(l+1)})}_{<0} + \sum_{\substack{j=k \\ \pi(j)\in N_0}}^{l} \underbrace{\beta_{\pi(j)\pi(l+1)}}_{\geq 0}\underbrace{(\hat{y}_{\pi(j)} - \hat{y}_{\pi(l+1)})}_{<0},
\end{aligned}
$$

o que implica $\Theta(\pi', \hat{N}_0, \hat{N}_1; x^*) \leq \Theta(\pi, \hat{N}_0, \hat{N}_1; x^*)$. Então π' também minimiza $\Theta(\pi, \hat{N}_0, \hat{N}_1; x^*)$ e ordena pelo menos mais uma entrada em $\hat{y}$. Repetindo o processo um número finito de passos, chegamos a uma permutação π que satisfaz (6). $\square$

Teorema 3. *Dado $x^* \in [0,1]^n$, a solução do problema de separação (5) ocorre em $(\hat{N}_0, \hat{N}_1)$ e $\hat{\pi} \in S_n$ tais que*

$$
\hat{N}_0 = \left\{ j \in N : \overline{x^*_j} > x^*_j \right\}, \ \hat{N}_1 = \left\{ j \in N : x^*_j \geq \overline{x^*_j} \right\},
$$

$$
\hat{y}_{\hat{\pi}(1)} \geq \hat{y}_{\hat{\pi}(2)} \geq \ldots \geq \hat{y}_{\hat{\pi}(n)},
$$

onde $\hat{y}_j = \overline{x^*_j}$, se $j \in \hat{N}_0$, e $\hat{y}_j = x^*_j$, se $j \in \hat{N}_1$.

Demonstração. Seja $(N_0, N_1) \in \mathcal{N}$, $\pi \in S_n$ uma solução ótima para (5). Pelo Lema 2, podemos admitir que π é tal que $y_{\pi(1)} \geq y_{\pi(2)} \geq \ldots \geq y_{\pi(n)}$, onde $y_j = \overline{x_j^*}$ se $j \in N_0$, $y_j = x_j^*$, se $j \in N_1$. Se $(N_0, N_1) = (\hat{N}_0, \hat{N}_1)$, obtemos o resultado desejado. Caso contrário, existe um inteiro $l \geq 1$ tal que

$$\pi(j) \in N_0 \Leftrightarrow \pi(j) \in \hat{N}_0 \quad \forall j = 1, \ldots, l-1 \quad \text{e}$$

$$\pi(l) \in N_0 \backslash \hat{N}_0 = N_0 \cap \hat{N}_1 \quad \text{ou} \quad \pi(l) \in N_1 \backslash \hat{N}_1 = N_1 \cap \hat{N}_0$$

Suponha primeiro que $\pi(l) \in N_0 \cap \hat{N}_1$. Defina $N_0' = N_0 \backslash \{\pi(l)\}$ e $N_1' = N_1 \cup \{\pi(l)\}$. Temos que $y_{\pi(l)} = \overline{x^*}_{\pi(l)} \leq x^*_{\pi(l)} = \overline{y}_{\pi(l)}$. Seja $k \leq l$ tal que

$$y_{\pi(1)} \geq \ldots \geq y_{\pi(k-1)} \geq \overline{y}_{\pi(l)} \geq y_{\pi(k)} \geq \ldots \geq y_{\pi(l-1)} \geq y_{\pi(l+1)} \geq \ldots \geq y_{\pi(n)}.$$

Pelo Lema 2, a ordenação π' que minimiza $\Theta(\pi, N_0', N_1'; x^*)$ é a que ordena os valores acima, ou seja,

$$\pi'(j) = \begin{cases} \pi(j), & j = 1 \ldots k-1, \text{ ou } i = l+1 \ldots n, \\ \pi(l), & j = k, \\ \pi(j-1), & j = k+1 \ldots l. \end{cases} \tag{7}$$

Seja $W = \Theta(\pi', N_0', N_1'; x^*) - \Theta(\pi, N_0, N_1; x^*)$. Então, $W = B + C$ tais que

$$B = \sum_{i=1}^{l-1} \alpha_{\pi(l)\pi(i)} (\overline{x^*}_{\pi(i)} + \overline{x^*}_{\pi(l)}) - \sum_{\substack{j=k \\ \pi(j) \in N_1}}^{l-1} \alpha_{\pi(l)\pi(j)} (\overline{x^*}_{\pi(j)} + \overline{x^*}_{\pi(l)}) - \sum_{\substack{j=k \\ \pi(j) \in N_0}}^{l-1} \alpha_{\pi(l)\pi(j)} (x^*_{\pi(j)} + x^*_{\pi(l)}),$$

$$C = \sum_{i=k}^{l-1} \beta_{\pi(l)\pi(i)} x^*_{\pi(i)} - \sum_{\substack{j=k \\ \pi(j) \in N_1}}^{l-1} \beta_{\pi(l)\pi(j)} x^*_{\pi(l)} - \sum_{\substack{j=k \\ \pi(j) \in N_0}}^{l-1} \beta_{\pi(j)\pi(l)} x^*_{\pi(j)}.$$

Portanto,

$$\begin{aligned} W &= \sum_{\substack{j=k \\ \pi(j) \in N_0}}^{l-1} \alpha_{\pi(l)\pi(j)} (\overline{x^*}_{\pi(j)} + \overline{x^*}_{\pi(l)} - x^*_{\pi(j)} - x^*_{\pi(l)}) + \sum_{\substack{j=k \\ \pi(j) \in N_1}}^{l-1} \beta_{\pi(j)\pi(l)} (x^*_{\pi(j)} - x^*_{\pi(l)}) \\ &= \sum_{\substack{j=k \\ \pi(j) \in N_0}}^{l-1} \alpha_{\pi(j)\pi(l)} 2(\underbrace{\overline{x^*}_{\pi(j)}}_{=y_{\pi(j)}} - \underbrace{x^*_{\pi(l)}}_{=\overline{y}_{\pi(l)}}) + \sum_{\substack{j=k \\ \pi(j) \in N_1}}^{l-1} \beta_{\pi(j)\pi(l)} (\underbrace{x^*_{\pi(j)}}_{=y_{\pi(j)}} - \underbrace{x^*_{\pi(l)}}_{=\overline{y}_{\pi(l)}}), \end{aligned}$$

levando a $\Theta(\pi', N_0', N_1'; x^*) \leq \Theta(\pi, N_0, N_1; x^*)$. Logo $(N_0', N_1') \in \mathcal{N}$, $\pi' \in S_n$ também é solução ótima e (N_0', N_1') coincide em $(\hat{N}_0, \hat{N}_1)$ em pelo menos mais um elemento (em relação a (N_0, N_1)). Repetindo o processo um número finito de passos, mostramos que $(\hat{N}_0, \hat{N}_1)$, $\hat{\pi} \in S_n$ é solução ótima.

Para o caso complementar, considere que $\pi(l) \in N_1 \cap \hat{N}_0$. De forma similar, vamos definir $N_0' = N_0 \cup \{\pi(l)\}$ e $N_1' = N_1 \setminus \{\pi(l)\}$. Agora, $y_{\pi(l)} = x^*_{\pi(l)} < \overline{x^*}_{\pi(l)} = \overline{y}_{\pi(l)}$. Seja $k \leq l$ tal que

$$y_{\pi(1)} \geq \ldots \geq y_{\pi(k-1)} \geq \overline{y}_{\pi(l)} > y_{\pi(k)} > \ldots \geq y_{\pi(l-1)} \geq y_{\pi(l+1)} \geq \ldots \geq y_{\pi(n)}.$$

Pelo Lema 2, esta ordenação define a permutação π', dada por (7), que minimiza $\Theta(\pi, N_0', N_1'; x^*)$. Neste caso, $W = \Theta(\pi', N_0', N_1'; x^*) - \Theta(\pi, N_0, N_1; x^*) = E + F$, onde

$$E = -\sum_{\substack{j=k \\ \pi(j) \in N_1}}^{l-1} \alpha_{\pi(l)\pi(j)} (\overline{x^*}_{\pi(l)} + \overline{x^*}_{\pi(j)}) + \sum_{i=k}^{l-1} \alpha_{\pi(l)\pi(i)} (x^*_{\pi(i)} + x^*_{\pi(l)}) - \sum_{\substack{j=k \\ \pi(j) \in N_0}}^{l-1} \alpha_{\pi(l)\pi(j)} (x^*_{\pi(l)} + x^*_{\pi(j)}),$$

$$F = -\sum_{\substack{i=k\\ \pi(j)\in N_1}}^{l-1} \beta_{\pi(l)\pi(j)} x^*_{\pi(l)} + \sum_{i=k}^{l-1} \beta_{\pi(l)\pi(j)} x^*_{\pi(l)} - \sum_{\substack{j=k\\ \pi(j)\in N_0}}^{l-1} \beta_{\pi(l)\pi(i)} x^*_{\pi(j)}.$$

Daí deduzimos que

$$\begin{aligned} W &= \sum_{\substack{j=k\\ \pi(j)\in N_1}}^{l-1} \alpha_{\pi(l)\pi(j)}(x^*_{\pi(l)} + x^*_{\pi(j)} - \overline{x^*}_{\pi(l)} - \overline{x^*}_{\pi(j)}) + \sum_{\substack{j=k\\ \pi(j)\in N_0}}^{l-1} \beta_{\pi(j)\pi(l)}(x^*_{\pi(l)} - x^*_{\pi(j)}) \\ &= \sum_{\substack{j=k\\ \pi(j)\in N_1}}^{l-1} \alpha_{\pi(l)\pi(j)} 2(\underbrace{\overline{x^*}_{\pi(j)}}_{=y_{\pi(j)}} - \underbrace{\overline{x^*}_{\pi(l)}}_{=\overline{y}_{\pi(l)}}) + \sum_{\substack{j=k\\ \pi(j)\in N_0}}^{l-1} \beta_{\pi(j)\pi(l)}(\underbrace{\overline{x^*}_{\pi(j)}}_{=y_{\pi(j)}} - \underbrace{\overline{x^*_{\pi(l)}}}_{=\overline{y}_{\pi(l)}}), \end{aligned}$$

ou ainda $\Theta(\pi', N'_0, N'_1; x^*) \leq \Theta(\pi, N'_0, N'_1; x^*)$. Novamente, obtemos uma solução ótima (N'_0, N'_1), $\pi \in S_n$ onde (N'_0, N'_1) coincide com $(\hat{N}_0, \hat{N}_1)$ em pelo menos mais um elemento, levando ao resultado desejado. □

Aplicação da t-Linearização ao Problema de Corte Máximo

O problema do Corte Máximo (Max-Cut) é um dos problemas de otimização combinatória que, devido a seu domínio de aplicações, tais como a física estatística, fluxos em redes, projeto de circuitos VLSI etc, tem atraído o interesse científico de pesquisadores das áreas de matemática discreta e otimização matemática [Deza e Laurent, 1997; Steven et al., 1998; Burer et al., 2001/2002; Helmberg et al., 1996]. Sendo um problema NP-Difífil [Karp, 1972], é considerado um desafio computacional resolvê-lo de forma ótima, mesmo para instâncias de tamanhos moderados [Krislock et al., 2014].

Esse problema pode ser descrito como segue. É dado um grafo não-direcionado $G = (V, E)$, com $|V| = n$ vértices e $|E| = m$ arestas ponderadas. Consideramos o peso $c_{ij} = c_{ji}$ para $\{i, j\} \in E$, e $c_{ij} = 0$ para $(i, j) \notin E$. Qualquer bipartição $(S, \overline{S} := V \backslash S)$ dos vértices define um corte de G, ou seja, um subconjunto de arestas com uma extremidade em S e a outra em $\overline{S}$, onde S ou $\overline{S}$ pode ser vazio. O problema Max-Cut consiste em encontrar um corte $(S, \overline{S})$ tal que a soma dos pesos de suas arestas seja máximo.

Se representarmos a bipartição $(S, \overline{S})$ pelo vetor binário $x \in \{0, 1\}^n$, com $x_i = 1$ se e somente se $i \in S$, então Max-Cut pode ser formulado como o seguinte problema quadrático sem restrições (onde subentende-se $\overline{x} = \mathbf{1} - x$):

$$(MC) \quad \max\left\{\sum_{i=1}^{n}\sum_{j=i+1}^{n} c_{ij}(x_i\overline{x}_j + \overline{x}_i x_j) : x \in \{0,1\}^n\right\}. \tag{8}$$

Observe que a função objetivo é

$$F(x) = \sum_i \sum_{j>i} c_{ij} x_i (1 - x_j) + \sum_i \sum_{j>i} c_{ij}(1 - x_i) x_j = \sum_i \sum_{j>i} (-2c_{ij}) x_i x_j + \sum_i \sum_j c_{ij} x_i.$$

Usando o desenvolvimento da seção anterior, reescrevemos Max-Cut como:

$$(MC)_t \quad \begin{aligned} &\max t + \sum_{i=1}^{n}\sum_{j=1}^{n}(-c_{ij})^- x_i + \sum_{i=1}^{n}\sum_{j=1}^{n}(c_{ij}) x_i = t + \sum_i \sum_j \min\{0, c_{ij}\} x_i \\ &\text{s.a:}\quad t \leq \sum_i \sum_{\substack{j<i\\ \pi(j)\in N_1}} c^+_{\pi(j)\pi(i)}(\overline{x}_{\pi(i)} + \overline{x}_{\pi(j)}) + \sum_i \sum_{\substack{j<i\\ \pi(j)\in N_0}} c^+_{\pi(j)\pi(i)}(x_{\pi(i)} + x_{\pi(j)}) + \\ &\qquad \sum_i \sum_{\substack{j<i\\ \pi(j)\in N_1}} -2c^-_{\pi(j)\pi(i)} x_{\pi(i)} + \sum_i \sum_{\substack{j<i\\ \pi(j)\in N_0}} -2c^-_{\pi(j)\pi(i)} x_{\pi(j)} \forall \pi \in S_n \ \forall (N_0, N_1) \in \mathcal{N}, \\ &\qquad x \in \mathbb{B}^n. \end{aligned}$$

Por outro lado, a linearização clássica gera o seguinte modelo equivalente:

$$(MC)_{LC} \quad \begin{array}{ll} \max & \sum_{i=1}^{n}\sum_{j=i+1}^{n}(-2c_{ij})z_{ij} + \sum_{i=1}^{n}\sum_{j=i+1}^{n} c_{ij}x_i \\ \text{s.a:} & z_{ij} \leq x_i,\ z_{ij} \leq x_j,\ z_{ij} \geq x_i + x_j - 1, \\ & x \in \mathbb{B}^n. \end{array}$$

Uma terceira formulação matemática bem conhecida para Max-Cut é dada por:

$$\max\left\{\sum_{i=1}^{n}\sum_{j=i+1}^{n} c_{ij}\frac{1-y_iy_j}{2} : y_i^2 = 1 \forall i = 1,\ldots,n\right\},$$

onde $y_i = 1$ se $i \in S$ e $y_i = -1$ se $i \in \overline{S}$. Observando que função objetivo pode ser reescrita como $\frac{1}{4}y^tQy$, onde Q é a matriz simétrica com entradas $q_{ii} = \sum_{j=1}^{n} c_{ij}$ e $q_{ij} = q_{ji} - c_{ij}\ \forall i \neq j$, obtemos o seguinte modelo de programação semidefinida:

$$(MC)_{SDP} \quad \max\left\{\frac{1}{4}\langle Q, Y\rangle : Y = yy^t, Y_{ii} = 1 \forall i = 1,\ldots,n\right\}.$$

A notação $\langle Q, Y\rangle$ expressa a soma dos elementos da diagonal de Q^tY e vale $\langle Q, Y\rangle = y^tQy$, quando $Y = yy^t$ é uma matriz (simétrica) de posto 1. A relaxação de $(MC)_{SDP}$ consiste em substituir $Y = yy^t$ pela condição de que Y seja simétrica e semidefinida positiva. Ver detalhes em [Goemans e Williamson, 1995].

Experimentos Computacionais

Os experimentos computacionais foram conduzidos para testar a qualidade do limite superior e o tempo fornecido pelas relaxações da t-linearização $(MC)_t$, da linearização clássica $(MC)_{LC}$ e da formulação de programação semidefinida $(MC)_{SDP}$ para o problema Max-Cut.

Todas as instâncias foram geradas a partir da combinação da densidade d de arestas do grafo ($d \in \{10, 25, 50 \text{ e } 75\%\}$), quantidade n de vértices ($n \in \{30,\ 50,\ 70 \text{ e } 100\}$) e custos c_{ij} das arestas ($c_{ij} \in [-100, 100], c \neq 0$). Para cada par de vértices (i, j), a aresta $\{i, j\}$ vai existir no grafo com probabilidade d e, caso ocorra, receberá um custo no intervalo.

Para a implementação da formulação de programação semidefinida, foi usado o software e a linguagem MATLAB© com adição da Toolbox de rápida prototipação de problemas de otimização YALMIP, em conjunto com o Solver SDPA (SemiDefinite Programming Algorithm). Já os modelos linearizados, foram implementados no Ambiente de Desenvolvimento Integrado (IDE - Integrated Development Environment) *Code::Blocks*, utilizando a linguagem C++. Os experimentos computacionais foram realizados em um processador Intel® Core™ i7 − 3770 com 3.40 GHz, 16 GB RAM e sistema operacional Windows 7 Professional para programação semidefinida ou sistema operacional Ubuntu 14.04 LTS para linearizações.

O experimento foi realizado da seguinte forma. Geramos 10 instâncias para cada combinação de n e d, contabilizando um total de 160. Para cada instância, calculamos a solução relaxada retornada por cada uma das três abordagens, o tempo usado para obter tal solução e o $gap = [\frac{\overline{w}-w^*}{w^*} \times 100]$, onde $\overline{w}$ é o valor da solução relaxada e w^* o valor da solução ótima. A Tabela 1 apresenta as comparações dos tempos médios de computação e dos gaps médios para cada grupo de 10 instâncias associadas a uma mesma combinação de n e d.

Observamos que as médias dos limites fornecidos pela t-linearização são melhores do que as médias dos limites obtidos pela linearização clássica, exceto para a combinação $n = 100$ e $d = 75\%$, onde a diferença, embora bem pequena, favorece à segunda. Para d fixo, os gaps médios crescem com o aumento do tamanho do grafo em ambas as abordagens. O mesmo ocorre quando fixamos n e aumentamos a densidade. Embora esse crescimento também ocorra na programação

Tabela 1: Média Gap(%) e Tempo(s)

d	n	**Médias**					
		Clássica		**SDP**		**t-linearização**	
		Tempo(s)	Gap(%)	Tempo(s)	Gap(%)	Tempo(s)	Gap(%)
10	30	0.03	14.77	0.97	0.00	0.01	4.21
	50	0.34	38.10	9.69	0.00	0.04	25.29
	70	0.20	66.32	61.17	0.00	0.09	56.45
	100	0.28	93.14	461.02	7.92	0.32	85.51
25	30	0.12	75.29	0.95	0.00	0.01	52.39
	50	0.19	128.31	9.81	2.85	0.03	112.41
	70	0.26	162.64	59.24	10.16	0.09	149.83
	100	0.22	205.82	450.32	11.27	0.36	196.37
50	30	0.25	135.54	1.02	0.00	0.01	111.55
	50	0.05	192.64	9.84	13.27	0.03	172.93
	70	0.31	242.88	58.30	9.35	0.09	229.83
	100	0.22	311.79	453.01	7.90	0.39	300.89
75	30	0.38	199.22	1.11	0.00	0.01	171.67
	50	0.06	269.60	10.22	8.45	0.03	250.41
	70	0.27	334.02	60.17	10.32	0.10	318.50
	100	0.26	403.78	462.50	5.30	0.38	404.65

semidefinida, ele é bem mais modesto. Na verdade, como esperado, essa abordagem destaca-se como a melhor opção para obtenção de bons limites.

Já os resultados com relação à média dos tempos mostram que as técnicas de linearização demandam esforço computacional bem menor que a programação semidefinida, com prevalência da t-linerarização até 75 vértices. Os tempos aumentam com o tamanho do grafo, especialmente na programação semidefinida, porém a variação é pouco significativa quando fixamos n e mudamos a densidade.

Conclusão

Nesse artigo, revisitamos a técnica da t-linearização, proposta por Rodrigues [2010]. Para um problema quadrático binário sem restrições onde os coeficientes dos termos quadráticos são não-negativos, a relaxação do problema t-linearizado fornece a solução ótima [Rodrigues et al., 2012]. Aqui generalizamos a técnica para aplicação em problemas quadráticos binários com coeficientes arbitrários na função objetivo. Mostramos como contruir um problema linear equivalente, com o acréscimo de uma única variável, mas um número exponencial de restrições. Demonstramos, por outro lado, como separar tais restrições em tempo polinomial.

Aplicamos a técnica ao problema do corte máximo, tendo em vista que o mesmo possuí uma formulação com função objetivo quadrática. A t-linearização se mostrou competitiva em comparação com a linearização clássica em termos de limites e tempo de computação. No entanto, essa abordagem precisa incoporar outros elementos para que se torne competitiva com relação á programação semidefinda.

Referências

Balas, E. (1964). Extension de l'algorithme additif à la programmation en nombres entiers et à la programmation non linéaire. *Comptes Rendus Hebdomadaires des Seances de L'Academie des Sciences*, 258(21):5136.

Billionnet, A. e Sutter, A. (1994). Minimization of a quadratic pseudo-boolean function. *European Journal of Operational Research*, 78(1):106–115.

Boros, E. e Hammer, P. L. (2002). Pseudo-boolean optimization. *Discrete Applied Mathematics*, 123(1):155–225.

Burer, S., Monteiro, R., e Zhang, Y. (2001/2002). Rank-two relaxation heuristic for max-cut and other binary quadratic problems. *Siam Journal on Optimization*, 12(2):503–521.

Chardaire, P. e Sutter, A. (1995). A decomposition method for quadratic zero-one programming. *Management Science*, 41(4):704–712.

Dantzig, G. B. (1963). Linear programming and extensions.

Deza, M. e Laurent, M. (1997). *Geometry of Cuts and Metrics: Algotithms and Combinatorics*. Springer, Berlin.

Elloumi, S., Faye, A., e Soutif, E. (2000). Decomposition and linearization for 0-1 quadratic programming. *Annals of Operations Research*, 99(1):79–93.

Fortet, R. (1960). L'algebre de Boole et ses applications en recherche operationnalle. *Tabajos de Estadistica*, 11(2):111–118.

Gary, M. R. e Johnson, D. S. (1979). Computers and intractability: A guide to the theory of NP-completeness.

Glover, F. (1975). Improved linear integer programming formulations of nonlinear integer problems. *Management Science*, 22(4):455–460.

Goemans, M. X. e Williamson, D. P. (1995). Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *Journal of the ACM (JACM)*, 42(6): 1115–1145.

Gueye, S. e Michelon, P. (2009). A linearization framework for unconstrained quadratic (0-1) problems. *Discrete Applied Mathematics*, 157(6):1255–1266.

Hammer, P. L. e Rudeanu, S. (1968). Boolean methods in operations research.

Helmberg, C., Rendl, F., Vanderbei, R., e Wolkowicz, H. (1996). An interior-ponit method for semidefinite programming. *Siam Journal on Optimization*, 6(2):342–361.

Hu, T. C. (1969). Integer programming and network flows. Technical report, DTIC Document.

Karp, R. M. (1972). *Reducibility among combinatorial problems*. Springer.

Krislock, N., Malick, J., e Rouoin, F. (2014). Improved semidefinite bounding procedure for solving max-cut problems to optimality. *Mathematical Programming*, 143(1–2):61–86.

Rendl, F., Rinaldi, G., e Wiegele, A. (2010). Solving max-cut to optimality by intersecting semidefinite and polyhedral relaxations. *Mathematical Programming*, 121(2):307.

Rodrigues, C. D. (2010). *Abordagens híbridas na solução de problemas de programação inteira da teoria e prática*. PhD thesis, Universidade Federal do Ceará e Université d'Avignon.

Rodrigues, C., Quadri, D., Michelon, P., e Gueye, S. (2012). 0-1 quadratic knapsack problems: an exact approach based on a t-linearization. *SIAM Journal on Optimization*, 22(4):1449–1468.

Steven, J. B., Yinyu., e Zhang, X. (1998). Solving large-scale sparse semidefinite programs for combinatorial optimization. *Siam Journal on Optimization*, 10:443–461.