

## Formulações para Árvore Geradora Mínima com Grau Mínimo e Centrais e Terminais Fixos

**Manoel Campêlo, Rafael Castro de Andrade**

Departamento de Estatística e Matemática Aplicada, Universidade Federal do Ceará  
Campus do Pici, Bloco 910, Pici, Fortaleza, Ceará, CEP 60440-900  
{mcampelo, rca}@lia.ufc.br

**Fábio Carlos Sousa Dias, Críston Pereira de Souza**

Curso de Ciência da Computação, Universidade Federal do Ceará  
Campus de Quixadá, Quixadá, Ceará, CEP 63900-000  
fabiocsd@lia.ufc.br, criston@ufc.br

### RESUMO

O Problema de Árvore Geradora Mínima com Restrição de Grau Mínimo e Centrais e Terminais Fixos - MDF-MST consiste em encontrar uma árvore geradora mínima de um grafo, onde os vértices são particionados em dois subconjuntos chamados centrais e terminais, de modo que cada vértice terminal seja folha da árvore e os vértices centrais satisfaçam uma restrição de grau mínimo. Neste trabalho propomos várias formulações de programação inteira para MDF-MST e comparamos teórica e computacionalmente a qualidade do limite inferior gerado por suas relaxações lineares. Desenvolvemos um algoritmo de planos-de-corte exato, cujo desempenho computacional se mostra superior aos resultados prévios da literatura para instâncias geradas aleatoriamente.

**PALAVRAS CHAVE.** Árvore geradora mínima com restrição de grau, Formulação matemática, Programação inteira, Método de planos-de-corte.

**Área Principal:** Otimização Combinatória, Programação Matemática

### ABSTRACT

The Min-degree Constrained Minimum Spanning Tree with Fixed Centrals and Terminals - MDF-MST consists in finding a minimum spanning tree of a graph, where the vertices are partitioned into centrals and terminals, such that each terminal is a leaf of the tree and each central satisfies a minimum-degree constraint. We propose several ILP formulations for MDF-MST. We theoretically compare the lower bounds given by the linear relaxations as well as we compare them by computational experiments. Also, we develop a cutting-plane algorithm whose computational performance shows to be superior to other algorithms from the literatura for random generated instances.

**KEYWORDS.** Degree-constrained minimum spanning tree, ILP formulation, Integer programming, Cutting-plane Method.

**Main Area:** Combinatorial Optimization, Mathematical Programming

## 1. Introdução

Uma árvore é um grafo minimamente conectado ou, mais precisamente, um grafo acíclico conexo. Uma árvore geradora de um grafo é uma árvore contendo todos os seus vértices. Se o grafo modela uma rede, então uma árvore geradora compreende um subconjunto minimal de links que permitem comunicação entre todos os pares de nós da rede; ou ainda, é uma subestrutura da rede que garante conectividade sem links redundantes. Existe uma ampla variedade de aplicações que podem ser modeladas por árvore geradora ou árvore geradora satisfazendo restrições adicionais.

Dado um grafo ponderado  $G = (V, E)$  com conjunto de vértices  $V$ , conjunto de arestas  $E$  e custo  $c_e$  associado com cada aresta  $e \in E$ , o custo da árvore geradora  $H = (V, E')$  é  $c(H) = \sum_{e \in E'} c_e$ . O problema básico neste contexto, a ser denotado MST, é encontrar uma árvore geradora de custo mínimo. Há diversas variantes deste problema fundamental, onde se impõem condições adicionais à desejada árvore geradora  $H$  (Achuthan et al., 1994; Gouveia, L., 1995; Gouveia, L. e Telhada, J., 2011). Uma dessas restrições adicionais está relacionada ao grau  $d_H(v)$  de cada vértice  $v \in V$ , isto é, ao número de arestas incidentes a  $v$  em  $H$  (Almeida et al., 2006; Narula, HO, 1980).

Dias et al. (2013) introduziram uma nova variante do MST, aqui chamado de *Problema da Árvore Geradora Mínima com Restrição de Grau Mínimo e Centrais e Terminais Fixos* (MDF-MST). Neste caso, o conjunto de vértices  $V$  é particionado em  $V = C \cup T$ ,  $C \cap T = \emptyset$ , onde os vértices em  $C$  e  $T$  são chamados de centrais e terminais, respectivamente.

**MDF-MST:** Dado um grafo simples, conexo,  $G = (C \cup T, E)$ , com custo  $c_e \in \mathbb{R}$ , associado a cada aresta  $e \in E$ , e valor de grau mínimo  $d_v \geq 1$ , relativo a cada vértice  $v \in C$ , deseja-se encontrar uma árvore geradora mínima  $H$  de  $G$ , onde  $d_H(v) \geq d_v$ , se  $v \in C$ , e  $d_H(v) = 1$ , se  $v \in T$ .

Este problema constitui-se numa variação do problema definido por Almeida, Martins e Sousa (2006), onde os centrais e terminais não são definidos a priori. Na próxima seção, propomos algumas formulações de programação inteira para MDF-MST. Nas Seções 3 e 4, comparamos teórica e computacionalmente os limites inferiores fornecidos pelas relaxações lineares destas formulações. Na Seção 5, propomos uma modificação do algoritmo de planos-de-corte de Dias et al. (2013) e comparamos o desempenho computacional dessas versões. Os testes foram realizados com instâncias da literatura, que se mostraram de fácil solução com o novo algoritmo. Em seguida, geramos novas instâncias e realizamos experimentos computacionais similares na Seção 6.

Ao longo do texto, adotamos a seguinte notação relativa a  $G$  e subconjuntos  $U, U' \subseteq V(G)$ .  $G(U)$  é o subgrafo induzido por  $U$  e  $\delta(U : U')$  são as arestas com uma extremidade em  $U$  e outra em  $U'$ . Por simplicidade, usamos  $E(U) = \delta(U : U)$ ,  $\delta(U) = \delta(U : V \setminus U)$  e  $\delta(\{v\}) = \delta(v)$ , para  $v \in V$ . Se atribuímos orientação às arestas de  $G$ , usamos a notação  $\delta^+(U : U')$  para nos referirmos aos arcos que saem de  $U$  e chegam a  $U'$ , e  $\delta^-(U : U') = \delta^+(U' : U)$ . A quantidade de centrais, de terminais e de vértices é  $c = |C|$ ,  $t = |T|$  e  $n = c + t$ . Quando o grau mínimo for comum a todos os vértices centrais, usaremos a letra  $d$  para representar esse valor comum.

## 2. Formulação Matemática

Uma solução viável para o MDF-MST consiste em uma árvore geradora entre os centrais, e ligações entre terminais e centrais, de modo a satisfazer as restrições de grau mínimo. Desse modo, nas formulações propostas, aplicamos as restrições de árvore geradora apenas ao subgrafo  $G(C)$  induzido pelos centrais. Restrições adicionais garantem a conectividade entre terminais e centrais, bem como a satisfação dos requisitos de grau mínimo. S.p.g, supomos  $\delta(T) = \emptyset$ .

Para definir uma árvore geradora de  $G(C)$ , consideramos as formulações baseadas em eliminação de subciclos - MST<sub>SEC</sub> (Dantzig et al., 1954), cortes direcionados - MST<sub>cutD</sub><sup>r</sup> (Magnanti e Wolsey, 1995), interseção de arestas - MST<sub>int</sub> (Martin, 1991), fluxo simples - MST<sub>floD</sub><sup>r</sup> (Almeida et al., 2006) ou restrições de Miller-Tucker-Zemlin - MST<sub>MTZ</sub><sup>r</sup> (Gouveia, 1995). As três primeiras fornecem poliedros inteiros; as duas últimas usam menos variáveis e restrições. Observamos que, para o problema onde os centrais e terminais não são definidos a priori, formulações baseadas em MST<sub>int</sub> e MST<sub>floD</sub><sup>r</sup> mostraram-se eficientes (Almeida et al., 2006; Martines e Cunha, 2012).

Para  $MST_{SEC}$  e  $MST_{int}$ , definimos as variáveis binárias  $x_e$  que indicam se aresta  $e \in E$  é escolhida para compor a solução. Para as outras formulações, convertamos o grafo  $G = (V, E)$  em um grafo direcionado  $D = (V, A)$ , substituindo cada aresta  $e = \{i, j\} \in E$  pelo arcos  $(i, j)$  e  $(j, i)$ , ambos com custo  $c_e$ . Usamos as variáveis binárias  $y_{ij}$  para indicar se o arco  $(i, j)$  está na solução. Nas formulações  $MDF_{cutD}^r$ ,  $MDF_{floD}^r$  e  $MDF_{MTZ}^r$ , um nó central  $r$  é escolhido arbitrariamente para ser a raiz da árvore, de modo que o caminho de  $r$  até  $j \in V \setminus \{r\}$  na solução seja orientado de  $r$  para  $j$ . Note que para cada raiz  $r$  escolhida temos uma formulação distinta. Por simplicidade, retiramos de  $A$  todos os arcos de  $T$  para  $C$  e todos os arcos de  $C$  para  $r$ , pois nunca participam da solução.

**Eliminação de subtours ( $MDF_{SEC}$ ):** As restrições de  $MST_{SEC}(G(C))$  são as equações (1) e (2). A Equação (3) estabelece o grau mínimo de cada central, e a Equação (4) o grau unitário de cada terminal. A Equação (5) indica as variáveis binárias.

$$\begin{aligned}
 (MDF_{SEC}) \quad & \min \sum_{e \in E} c_e x_e \\
 \text{sujeito a:} \quad & \sum_{e \in E(C)} x_e = c - 1 & (1) \\
 & \sum_{e \in E(S)} x_e \leq |S| - 1, & \forall S \subsetneq C \text{ com } |S| \geq 2 & (2) \\
 & \sum_{e \in \delta(i)} x_e \geq d_i, & \forall i \in C & (3) \\
 & \sum_{e \in \delta(t) \cap \delta(C)} x_e = 1, & \forall t \in T & (4) \\
 & x_e \in \{0, 1\}, & \forall e \in E & (5)
 \end{aligned}$$

**Coincidência de arestas ( $MDF_{int}$ ):** As restrições de  $MST_{int}(G(C))$  são as equações (6)–(9). A Equação (10) estabelece o grau mínimo de cada central, e a Equação (11) o grau unitário de cada terminal. As equações (12) e (13) indicam as variáveis binárias.

$$\begin{aligned}
 (MDF_{int}) \quad & \min \sum_{e \in E(C)} c_e x_e \\
 \text{sujeito a:} \quad & \sum_{e \in E(C)} x_e = c - 1 & (6) \\
 & x_e = \lambda_{ij}^k + \lambda_{ji}^k, & \forall k \in C, \forall e = \{i, j\} \in E(C) & (7) \\
 & \sum_{(i,j) \in \delta(j) \cap E(C)} \lambda_{ij}^k = 1, & \forall j \in C, \forall k \in C, k \neq j & (8) \\
 & \lambda_{ik}^k = 0, & \forall i \in C, \forall k \in C \cap \delta(i) & (9) \\
 & \sum_{e \in \delta(i)} x_e \geq d_i, & \forall i \in C & (10) \\
 & \sum_{e \in \delta(t) \cap \delta(C)} x_e = 1, & \forall t \in T & (11) \\
 & \lambda_{ij}^k, \lambda_{ji}^k \in \{0, 1\}, & \forall \{i, j\} \in E(C), \forall k \in C & (12) \\
 & x_e \in \{0, 1\}, & \forall e \in E & (13)
 \end{aligned}$$

**Cutsets direcionados ( $MDF_{cutD}^r$ ):** As restrições de  $MST_{cutD}^r(G(C))$  são as equações (14), (15) e (16). As equações (17) e (18) estabelecem o grau mínimo de cada central, e a Equação (19) o grau unitário de cada terminal. A Equação (20) indica as variáveis binárias.

$$\begin{aligned}
 (MDF_{cutD}^r) \quad & \min \sum_{(i,j) \in A} c_{ij} y_{ij} \\
 \text{sujeito a:} \quad & \sum_{(i,j) \in A(C)} y_{ij} = c - 1 & (14) \\
 & \sum_{(i,j) \in \delta^+(S: C \setminus S)} y_{ij} \geq 1, & \forall S \subsetneq C \text{ e } r \in S & (15) \\
 & \sum_{(i,j) \in \delta^-(j) \cap A(C)} y_{ij} = 1, & \forall j \in C \setminus \{r\} & (16) \\
 & \sum_{(i,j) \in \delta^+(i)} y_{ij} \geq d_i - 1, & \forall i \in C \setminus \{r\} & (17) \\
 & \sum_{(i,j) \in \delta^+(r)} y_{ij} \geq d_r & (18) \\
 & \sum_{(i,t) \in \delta^-(t) \cap \delta^+(C)} y_{it} = 1, & \forall t \in T & (19) \\
 & y_{ij} \in \{0, 1\}, & \forall (i, j) \in A & (20)
 \end{aligned}$$

Note que (16) e (19) podem ser unificadas como  $\sum_{(i,j) \in \delta^-(j)} y_{ij} = 1, \forall j \in V \setminus \{r\}$ .

**Fluxo simples orientado (MDF<sub>flOD</sub><sup>r</sup>):** As restrições de MST<sub>flOD</sub><sup>r</sup>(G(C)) são as equações (21), (22) e (23) para  $j \in C$ . As equações (24) e (25) estabelecem o grau mínimo de cada central; e (23) o grau unitário de cada terminal  $j \in T$ . As equações (26) e (27) indicam o domínio das variáveis.

$$(\text{MDF}_{\text{flOD}}^r) \min \sum_{(i,j) \in A} c_{ij} y_{ij}$$

$$\text{sujeito a: } \sum_{(j,i) \in \delta^-(i)} f_{ji} - \sum_{(i,j) \in \delta^+(i)} f_{ij} = 1, \quad \forall i \in C \setminus \{r\} \quad (21)$$

$$y_{ij} \leq f_{ij} \leq (c-1)y_{ij}, \quad \forall (i,j) \in A(C) \quad (22)$$

$$\sum_{(i,j) \in \delta^-(j)} y_{ij} = 1, \quad \forall j \in V \setminus \{r\} \quad (23)$$

$$\sum_{(i,j) \in \delta^+(i)} y_{ij} \geq d_i - 1, \quad \forall i \in C \setminus \{r\} \quad (24)$$

$$\sum_{(i,j) \in \delta^+(r)} y_{ij} \geq d_r \quad (25)$$

$$y_{ij} \in \{0, 1\}, \quad \forall (i,j) \in A \quad (26)$$

$$f_{ij} \geq 0, \quad \forall (i,j) \in A(C) \quad (27)$$

**Rótulos nos vértices (MDF<sub>MTZ</sub><sup>r</sup>):** As restrições de MST<sub>MTZ</sub><sup>r</sup>(G(C)) são as equações (28)–(30) e (31) para  $j \in C$ . As equações (32) e (33) estabelecem o grau mínimo de cada central, e a Equação (31) o grau unitário de cada terminal  $j \in T$ . A Equação (34) indica as variáveis binárias.

$$(\text{MDF}_{\text{MTZ}}^r) \min \sum_{(i,j) \in A} c_{ij} y_{ij}$$

$$\text{sujeito a: } u_i - u_j + c y_{ij} \leq c - 1, \quad \forall (i,j) \in A(C), j \neq r \quad (28)$$

$$1 \leq u_i \leq c - 1, \quad \forall i \in C \setminus \{r\} \quad (29)$$

$$u_r = 0 \quad (30)$$

$$\sum_{(i,j) \in \delta^-(j)} y_{ij} = 1, \quad \forall j \in V \setminus \{r\} \quad (31)$$

$$\sum_{(i,j) \in \delta^+(i)} y_{ij} \geq d_i - 1, \quad \forall i \in C \setminus \{r\} \quad (32)$$

$$\sum_{(i,j) \in \delta^+(r)} y_{ij} \geq d_r \quad (33)$$

$$y_{ij} \in \{0, 1\}, \quad \forall (i,j) \in A \quad (34)$$

As formulações de árvore geradora MST<sub>SEC</sub><sup>r</sup> e MST<sub>cutD</sub><sup>r</sup> usam um número linear de variáveis e possuem a propriedade de integralidade (pontos extremos inteiros). Entretanto, ambas apresentam um número exponencial de restrições, que em geral precisam ser separadas. Embora a separação em ambos os casos seja polinomial, este processo pode tornar-se computacionalmente dispendioso. A formulação MST<sub>int</sub><sup>r</sup> possui a propriedade de integralidade e uma quantidade de variáveis e restrições polinomial no tamanho da entrada. MST<sub>flOD</sub><sup>r</sup> possui quantidade polinomial de variáveis e restrições, porém o poliedro relaxado não é inteiro e a qualidade da solução relaxada depende da escolha da raiz  $r$ . Já a formulação MST<sub>MTZ</sub><sup>r</sup> elimina ciclos com um número linear de restrições, acrescentando apenas uma variável contínua para cada vértice, porém também não possui a propriedade de integralidade e é dependente da escolha da raiz.

### 3. Comparação Teórica das Relaxações Lineares

Nesta seção comparamos os limites inferiores gerados pelas formulações MDF<sub>SEC</sub><sup>r</sup>, MDF<sub>int</sub><sup>r</sup>, MDF<sub>cutD</sub><sup>r</sup>, MDF<sub>flOD</sub><sup>r</sup> e MDF<sub>MTZ</sub><sup>r</sup>. Utilizamos a comparação fornecida pela Proposição 1 para os poliedros das formulações do MST que serviram de base. Para as formulações onde as variáveis  $x$  não aparecem explicitamente no modelo, consideramos que a relação  $x_e = y_{ij} + y_{ji} \in \{0, 1\}$  foi acrescentada para cada aresta  $e = \{i, j\}$ . Dados uma formulação  $F$  (para MST ou MDF–MST) e um grafo  $G$ , denotamos por  $P(F, G)$  a projeção sobre  $x$  do conjunto viável relaxado da formulação  $F$  quando aplicada ao grafo  $G$ . Omitimos o grafo  $G$  quando se tratar do grafo de entrada do MDF.

**Proposição 1** (Magnanti e Wolsey, 1995; Martin, 1991) Para todo grafo  $H$  e todo  $r \in V(H)$ ,  $P(\text{MST}_{\text{SEC}}^r, H) = P(\text{MST}_{\text{cutD}}^r, H) = P(\text{MST}_{\text{int}}^r, H) \subseteq P(\text{MST}_{\text{flOD}}^r, H)$  e  $P(\text{MST}_{\text{SEC}}^r, H) \subseteq P(\text{MST}_{\text{MTZ}}^r, H)$ .

**Proposição 2** Seja  $PL(F)$  o valor ótimo da relaxação linear da formulação  $F$ . Para todo  $r \in C$ , temos que

1.  $PL(MDF_{cutD}^r) = PL(MDF_{SEC}) = PL(MDF_{int})$ ;
2.  $PL(MDF_{cutD}^r) \geq PL(MDF_{floD}^r)$ ;
3.  $PL(MDF_{cutD}^r) \geq PL(MDF_{MTZ}^r)$ ;

**Prova.** Da Proposição 1 temos que  $P(MST_{int}, G(C)) = P(MST_{SEC}, G(C))$ . Além disso, as restrições de grau em  $MDF_{SEC}$  e  $MDF_{int}$  são idênticas tanto para centrais quanto terminais (todas escritas na variável  $x$ ). Daí concluímos que  $P(MDF_{SEC}) = P(MDF_{int})$ . De forma análoga, pela Proposição 1, temos que  $P(MST_{cutD}^r, G(C)) \subseteq P(MST_{floD}^r, G(C))$  e  $P(MST_{cutD}^r, G(C)) \subseteq P(MST_{MTZ}^r, G(C))$  para todo  $r \in C$ . Como as restrições de grau nessas três formulações são idênticas (todas escritas na variável  $y$ ), obtemos os itens 2 e 3. Resta mostrar que  $PL(MDF_{cutD}^r) = PL(MDF_{SEC})$ . Para isso vamos mostrar que  $P(MDF_{cutD}^r) = P(MDF_{SEC})$ .

Seja  $x \in P(MDF_{cutD}^r)$ . Tome  $y$  tal que  $(x, y)$  é viável para a relaxação linear de  $MDF_{cutD}^r$ . Então  $x_e = y_{ij} + y_{ji}$ ,  $\forall e = \{i, j\} \in E$ ,  $\sum_{j \in \delta^+(r)} y_{rj} \geq d_r$ ,  $\sum_{j \in \delta^+(i)} y_{ij} \geq d_i - 1$  e  $\sum_{j \in \delta^-(i)} y_{ji} = 1$ , para todo  $i \in C \setminus \{r\}$ . Logo, temos que  $\sum_{e \in \delta(i)} x_e = \sum_{j \in \delta^+(i)} y_{ij} + \sum_{j \in \delta^-(i)} y_{ji} \geq d_i$ ,  $\forall i \in V$ . Além disso, como  $x \in P(MST_{cutD}^r, G(C)) = P(MST_{SEC}, G(C))$  pela Proposição 1, concluímos que  $x \in P(MDF_{SEC})$ .

Suponha agora  $x \in P(MDF_{SEC})$ . Denote por  $x^C$  e  $x^T$  os subvetores de  $x$  indexados por  $E(C)$  and  $\delta(C : T)$ , respectivamente. Então  $x^C \in P(MST_{SEC}, G(C))$ ,

$$\sum_{e \in \delta(i) \cap E(C)} x_e^C + \sum_{e \in \delta(i) \cap \delta(C:T)} x_e^T \geq d_i, \forall i \in C \quad \text{e} \quad \sum_{e \in \delta(t) \cap \delta(C)} x_e^T = 1, \forall t \in T. \quad (35)$$

Como  $P_{SEC}(G(C)) = P_{cutD}^r(G(C))$ , existe  $y^C$  tal que  $(x^C, y^C)$  satisfaz  $x_e^C = y_{ij}^C + y_{ji}^C$ ,  $\forall e = \{i, j\} \in E(C)$ ,  $\sum_{(i,j) \in A(C)} y_{ij}^C = c - 1$ , e  $\sum_{(i,j) \in \delta^+(S:C \setminus S)} y_{ij}^C \geq 1$ ,  $\forall S \subsetneq C$ ,  $r \in S$ . Logo  $y^C$  satisfaz (14) e (15). Tomando  $S = C \setminus \{j\}$ ,  $j \neq r$ , obtemos  $\sum_{i \in \delta^-(j)} y_{ij}^C = \sum_{(i,j) \in \delta^+(C \setminus \{j\} : \{j\})} y_{ij}^C \geq 1$ ,  $\forall j \in C \setminus \{r\}$ . Podemos mostrar que estas desigualdades são, na verdade, igualdades. Do contrário, somando-as obteríamos

$$\sum_{(i,j) \in A(C)} y_{ij}^C \geq \sum_{j \in C \setminus \{r\}} \sum_{i \in \delta^-(j)} y_{ij}^C > c - 1.$$

Na verdade, a igualdade que deve ocorrer ao longo da expressão acima mostra que

$$\sum_{i \in \delta^-(j)} y_{ij}^C = 1, \forall j \in C \setminus \{r\} \quad \text{e} \quad \sum_{i \in \delta^-(r)} y_{ir}^C = 0. \quad (36)$$

Logo,  $y^C$  satisfaz (19). Defina  $y = (y^C, x^T)$ . Então, por (35)-(36), concluímos que  $y$  satisfaz (16)–(18). Logo,  $x \in P(MDF_{cutD}^r)$ . ■

Mostraremos a seguir, através de um exemplo, que o limite da relaxação linear de qualquer dessas formulações pode ficar muito distante do ótimo. Considere o grafo da Figura 1(a), onde as arestas têm custo 0 ou  $M$ , conforme indicado. Sejam  $c_1, c_2, c_3, c_4$  os centrais e  $t_1, t_2$  os terminais. Considere  $d = 2$  e  $r = c_1$ . É fácil perceber que as duas arestas de peso  $M$  devem estar em qualquer solução viável, devido às restrições de grau. Sendo assim, uma solução ótima tem custo  $2M$ . Por outro lado, uma solução ótima relaxada para  $MDF_{cutD}^r$  é apresentada na Figura 1(b). O custo dessa solução é  $M$ , de modo que o gap de integralidade é  $M$ . Pela simetria do grafo, note que esse resultado independe da escolha da raiz.

Observamos que a Figura 1(a) ilustra uma situação incomum para instâncias de  $MDF$ – $MST$ . Na próxima seção avaliamos a qualidade da relaxação linear em instâncias aleatórias.

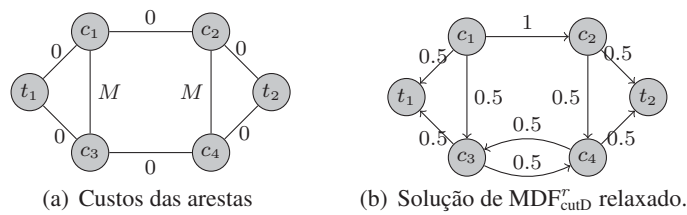


Figura 1: Instância de MDF-MST.

#### 4. Avaliação Computacional das Relaxações Lineares

Nesta seção, fazemos uma avaliação computacional da qualidade dos limites de programação linear, bem como dos tempos gastos para obtê-los, utilizando instâncias de Dias et al. (2013).

##### 4.1. Algoritmo de Plano de Corte

Nas formulações  $MDF_{SEC}$  e  $MDF_{cutD}^r$ , temos um conjunto de restrições de tamanho exponencial. Nesses casos, resolver a formulação ou mesmo sua relaxação linear com todas as restrições se torna impraticável. Uma alternativa é utilizar o método de planos de corte.

Para  $MDF_{cutD}^r$ , utilizamos uma modificação do algoritmo de planos de corte apresentado por Dias et al. (2013). Naquele algoritmo, a cada iteração adiciona-se a primeira restrição violada encontrada. Aqui, a cada iteração, adicionamos todas as restrições violadas, evitando a inclusão de duas restrições iguais. A política de remoção de restrições já adicionadas cujas folgas são maiores que uma certa constante  $\epsilon$  foi mantida em nosso algoritmo, com o mesmo valor para  $\epsilon = 0.5$ .

Na implementação do algoritmo de planos de corte para a  $MDF_{SEC}$ , verificamos que a política de adicionar todas as restrições violadas encontradas é muito mais rápida, comparada com a política de adicionar apenas uma. Além disso, evitar a inclusão de restrições iguais não acarretou redução de tempo computacional. Verificamos também que a política de remover restrições inativas não trouxe melhorias e ainda causou problemas de convergência. Dessa forma, para a  $MDF_{SEC}$  não estamos removendo restrições e nem verificando se há restrições idênticas na mesma iteração.

##### 4.2. Resultados Computacionais

Os algoritmos deste trabalho foram implementados e executados nas mesmas condições de Dias et. al. (2013), ou seja, foram implementados nas linguagens C/C++ e executados no sistema operacional Linux com um computador Intel Core i5, com 2.53 GHz e 4 GBytes de memória RAM. Utilizamos o Solver Matemático CPLEX na versão 11.1, através da biblioteca Concert Technology. Em todos eles, o tempo máximo de execução estabelecido foi de 9000 segundos (2,5 horas). O tempo de execução apresentado aqui corresponde ao tempo real (de relógio) decorrido.

Utilizamos as mesmas instâncias de teste que Dias et. al. (2013), derivadas das instâncias ALM de Martins e Souza (2009), definidas por grafos completos com custos positivos nas arestas. Nessas instâncias o grau mínimo é comum a todos os vértice, sendo  $d = 3$ .

Fizemos um comparativo entre as relaxações linear das 5 formulações. Utilizamos sempre como raiz o vértice de menor índice. Essa escolha baseou-se em experimentos computacionais preliminares, que mostraram pouca influência da raiz nos tempos de computação para essas instâncias.

Verificamos que os limites da relaxação linear para as formulações  $MDF_{SEC}$ ,  $MDF_{cutD}^r$  e  $MDF_{int}$  são muito fortes para as 147 instâncias testadas, o que se deve ao fato de serem tais formulações baseadas em restrições que descrevem completamente o politopo de árvores geradoras. Na verdade, uma grande quantidade de soluções da relaxação linear já são inteiras (em 96 das 147 instâncias), e mesmo aquelas soluções não-inteiras têm um valor muito próximo ao do ótimo inteiro. Ressalvamos que em alguns casos não foi possível resolver a relaxação linear de  $MDF_{int}$  por falta de memória, mas sabemos, pela Proposição 2, que seu valor é igual aos demais.

Já para as formulações  $MDF_{MTZ}^r$  e  $MDF_{floD}^r$ , nenhuma solução relaxada foi inteira. Os limites fornecidos por  $MDF_{floD}^r$  são quase sempre piores que aqueles de  $MDF_{MTZ}^r$  e significativamente

piores que os das outras três formulações. Os limites fornecidos por  $MDF_{MTZ}^r$  se aproximam um pouco mais, embora ainda fiquem distantes.

Um resumo desses resultados é apresentado na Tabela 1. Calculamos o gap relativo pela fórmula  $(OPT-LP)/OPT$ , onde OPT e LP são, respectivamente, o valor da solução ótima do problema e da relaxação linear. Os valores individuais para as instâncias são mostrados na Tabela 2. Os três dígitos após ALM indicam o valor de  $n$ . Por questão de espaço, mostramos apenas as instâncias cuja solução da relaxação linear das formulações  $MDF_{cutD}^r$ ,  $MDF_{SEC}$  e  $MDF_{int}$  não é inteira. Para cada formulação temos uma coluna com o valor e o tempo gasto (em segundos). O sinal “-” marca os casos onde houve falta de memória. Na última coluna, temos o valor ótimo da instância.

Formulações	Média GAP	Desvio Padrão GAP	Média Tempo (s)
CutsetD <sup>r</sup>	0,000219032	0,000603186	66,38
Subtour	0,000219032	0,000603186	72,997
Inter	0,000219032	0,000603186	748,66
MTZ <sup>r</sup>	0,040363461	0,026774423	2,9
FlowD <sup>r</sup>	0,070690925	0,022287836	52,34

Tabela 1: Média e desvio padrão dos GAP e média dos tempos

Com relação ao custo computacional, verificamos que a formulação  $MDF_{int}$  requer um tempo de computação e consumo de memória bem superiores. As formulações  $MDF_{SEC}$  e  $MDF_{cutD}^r$  são similares. Percebemos que, para instâncias onde a quantidade de centrais não é o limite máximo permitido para viabilidade, a formulação  $MDF_{SEC}$  é mais rápida; já quando a quantidade de centrais é exatamente a quantidade máxima permitida, a formulação  $MDF_{cutD}^r$  é melhor, exceto em 4 instâncias onde ela foi ligeiramente pior. Os tempos computacionais de  $MDF_{floD}^r$  são comparáveis àqueles dispendidos por  $MDF_{SEC}$  e  $MDF_{cutD}^r$ . Os tempos demandados por  $MDF_{MTZ}^r$  são os menores.

## 5. Método Exato de Solução

Nosso algoritmo para resolver o problema inteiro usa uma das formulações  $MDF_{cutD}^r$  ou  $MDF_{SEC}$  e emprega um procedimento de planos de corte semelhante àquele usado para resolver a relaxação linear. Avaliamos duas estratégias: (i) consideramos as restrições de integralidade desde o início, resolvendo a cada iteração um subproblema com variáveis inteiras, obtido daquele da iteração anterior, acrescido dos cortes encontrados ou (ii) primeiro resolvemos a relaxação linear, e só depois incluímos as restrições de integralidade, usando em ambas as etapas os planos de corte encontrados. Depois de alguns experimentos computacionais, verificamos que a segunda estratégia foi a mais promissora, mas que a política de remoção de restrições na segunda etapa não traz benefícios.

Sendo assim, nosso algoritmo exato usando o CPLEX executa em duas etapas. Na primeira, executamos o algoritmo de planos de corte descrito na Seção 4.1, com as mesmas políticas de geração, inserção e remoção de cortes. Caso a solução obtida não seja inteira, passamos à segunda etapa, quando convertemos as variáveis para binárias. Novamente, usamos o referido algoritmo de planos de corte, com a mesma política de inserção de cortes, mas sem remover restrições.

Depois de experimentos computacionais, não pudemos inferir claramente qual das duas formulações,  $MDF_{cutD}^r$  ou  $MDF_{SEC}$ , é a melhor. Optamos por utilizar a formulação  $MDF_{cutD}^r$ .

### 5.1. Experimentos Computacionais

Comparamos os tempos computacionais do nosso algoritmo e daqueles apresentados em Dias et. al. (2013). Na Tabela 3, temos a média dos tempos para instâncias com mesma quantidade de vértices. Nas colunas V1 e V2, encontram-se os resultados para as duas versões do algoritmo de Dias et al. (2013). Na coluna V3 estão os resultados para o algoritmo deste trabalho. O algoritmo V1 não encontrou a solução ótima de 28 instâncias dos grupos com 400 e 500 vértices, por falta de memória ou por ultrapassar o tempo máximo permitido. O mesmo aconteceu com o algoritmo V2 para 14 instâncias desses grupos. Considerando as instâncias resolvidas por ambos, a versão V2 obteve menor tempo. Já o algoritmo V3 encontra a solução ótima para todas as instâncias testadas. Podemos verificar que ele foi nitidamente superior a V1 e V2.

Os valores individuais para as instâncias são mostrados na Tabela 4. Nas colunas T1 e T2 estão os tempos dos algoritmos de Dias et al.(2013). Na coluna T3 estão os tempos do algoritmo

Problema	<i>c</i>	CutsetD	T(s)	Subtour	T(s)	MTZ	T(s)	Inter	T(s)	FlowD <sup>r</sup>	T(s)	<i>OPT</i>
ALM100-1	49	5756	0,16	5756	0,32	5461,51	0,14	5756	28,52	5508,39	0,4	5777
ALM100-2	40	5617	0,2	5617	0,12	5362,84	0,06	5617	4,74	5101,24	0,16	5639
	49	5673	0,26	5673	0,24	5145,6	0,14	5673	32,7	5263,65	0,46	5676
ALM200-1	90	7054,5	5,06	7054,5	3,06	6706,66	0,74	7054,5	341,64	6414,04	4,9	7055
ALM200-2	80	7842,5	3,74	7842,5	1,64	7351,79	0,52	7842,5	528,92	6976,52	2,88	7843
	99	8328	5,84	8328	16,02	7173,2	1,04	8328	679,22	7566,5	9,78	8331
ALM200-3	80	7630	4,08	7630	1,26	7276,34	0,52	7630	371,46	6819,38	2,24	7630
	90	7480,5	6,24	7480,5	2,1	7059,14	0,86	7480,5	710,8	6670,07	3,62	7482
ALM300-1	99	7698	8,08	7698	7,32	7185,61	0,92	7698	1075,2	6930,33	8,64	7723
	90	10377,5	6,06	10377,5	2,2	10138,05	0,72	10377,5	117,04	9738,71	4,24	10378
ALM300-2	100	9795,5	10,72	9795,5	5,82	9501,25	0,86	9795,5	384,2	9052,58	6,06	9796
	130	9002,5	33,3	9002,5	17,46	8502,59	2,1	9002,5	1604,92	8039,64	20,02	9006
	140	8979	36,04	8979	30,06	8409,66	2,92	8979	5698	8077,14	39,48	8986
	149	9298	51,16	9298	51,8	8516,63	3,72	-	-	8469,03	58,32	9317
ALM300-3	140	8419	40,44	8419	21,3	7982,84	2,86	8419	382,03	7607,36	35,42	8420
	149	8683	42,2	8683	38,4	8162,55	4,04	-	-	7925,31	59,02	8696
ALM300-3	110	9069,5	16,52	9069,5	9,48	8775,06	1,14	9069,5	621,1	8277,23	9,84	9070
	120	9097,5	30,26	9097,5	10,58	8709,66	1,6	9097,5	1232	8173,62	12,9	9098
	130	9049,5	39,7	9049,5	13,18	8614,23	2,38	9049,5	1337,64	8170,34	26,48	9051
	149	9788,5	60,58	9788,5	65,76	8753,75	3,34	9788,5	2028,07	9066,15	56,98	9800
ALM400-1	160	10316	71,56	10316	18,64	9852,76	4,52	10316	2830,61	9541,64	50,64	10322
	180	10070,25	111,46	10070,25	75,38	9553,8	7,22	-	-	9212,12	100,56	10075
	190	10455,67	163,3	10455,67	158,12	9640,8	8,46	-	-	9682,35	171,72	10459
	199	11442,5	167,46	11442,5	493,26	9814,45	8,88	-	-	10827,16	205,7	11457
ALM400-2	100	12739	15,96	12739	2,12	12423,25	0,84	12739	621,38	11982,66	6,52	12740
	110	12361	21,46	12361	2,78	12019,44	1,16	12361	1061,32	11559,99	10,46	12362
	120	11837	34,52	11837	5,72	11494,91	1,5	11837	759,22	10999,36	13,98	11838
	130	11391	50,28	11391	9,82	11041,67	1,94	11391	1464,22	10439,9	23,32	11393
	140	11000	85,06	11000	17,3	10651,3	2,78	11000	2663	9938,55	30,06	11002
	150	10856	85,24	10856	27,78	10538	3,48	10856	2533	9815,44	38,86	10858
	160	10713	132,28	10713	26,54	10347,59	4,66	10713	2007,97	9632,6	54,18	10717
	170	10634	146,24	10634	51,16	10180,53	5,92	-	-	9529,55	67,68	10637
	180	10676	159	10676	69,42	10127,02	7,32	-	-	9529,82	97,1	10680
	190	10918,5	240,4	10918,5	129,82	10181,14	10,04	-	-	9756,58	120,08	10923
199	11336,5	267,2	11336,5	251,98	10341,61	9,6	-	-	10389,81	213,18	11340	
ALM400-3	120	11932,5	25	11932,5	8,3	11516,48	1,48	11932,5	1605,38	11044,37	13,98	11933
	140	11484,5	41,64	11484,5	24	11088,16	3,1	11484,5	3042	10624,39	33,12	11486
	150	10984	60,18	10984	35,9	10592,75	3,76	10984	672,91	10013,47	40,26	10985
	160	10725	83,74	10725	39,4	10312,23	4,54	10725	7196	9764,96	59	10726
	180	10511	126,24	10511	126	9915,2	7	-	-	9565,84	109,44	10511
	190	10742	119,38	10742	265,78	10023,99	8,52	-	-	9882,13	139,2	10746
199	11440,5	169,92	11440,5	678,64	9957,1	10,08	-	-	10587,96	181,2	11444	
ALM500-1	220	11337	461,46	11337	349,12	10724,53	13,32	-	-	10164,53	244,82	11338
	240	11393	633,54	11393	406,64	10632,07	16,98	-	-	10249,5	437,26	11401
249	11816,12	653,62	11816,12	1146,88	10725,12	18,96	-	-	10892,96	647,44	11838	
ALM500-2	220	11456	348,44	11456	226,8	11008,12	13,18	-	-	10554,63	254,72	11461
	240	11741	412,26	11741	457,02	10981,66	16,88	-	-	10848,5	449,38	11753
249	12264,5	417,88	12264,5	1117	11037,94	19	-	-	11427,66	555,14	12266	
ALM500-3	200	11091	328,24	11091	108,46	10631,57	8,5	-	-	10010,61	147,68	11096
	220	11115,5	370,06	11115,5	212,64	10583,87	13,12	-	-	10015,14	241	11127
	240	11685,5	414,16	11685,5	586,36	10799,23	16,76	-	-	10715,42	475,54	11690
	249	12666,5	539,68	12666,5	2323,32	10879,92	19,56	-	-	11710,75	633,92	12686

Tabela 2: Comparação dos limites de Programação Linear para instâncias de Teste ALM



desde trabalho. Por questão de espaço, apresentamos apenas os resultados para as instâncias ALM de 500 vértices, grupo de instâncias onde o algoritmo de Dias et al.(2013) não encontrou a solução ótima em 13 casos. O sinal “-” marca os casos onde houve falta de memória e o sinal “\*” marca os casos onde o tempo máximo de 9000 segundos foi alcançado. As colunas OPT indicam o valor da solução ótima. Verificamos que o tempo computacional exigido por nosso algoritmo é nitidamente menor que para os dois outros. Mesmo quando o algoritmo V2 não encontra a solução ótima, nosso algoritmo é capaz de fazê-lo em tempo computacional aceitável.

Instâncias	V1 (Dias et al., 2013)	V2 (Dias et al., 2013)	V3
$n = 100$	0,27	0,25	0,15
$n = 200$	21,13	25,53	3,25
$n = 300$	450,42	325,02	21,28
$n = 400$	947,64*	1373,75*	65,28
$n = 500$	929,58*	1533,91*	179,24

Tabela 3: Média dos Tempos. (\*) Média considerando apenas as instâncias resolvidas

c	Tb8ch500_0				Tb8ch500_1				Tb8ch500_2			
	OPT	T1(s)	T2(s)	T3(s)	OPT	T1(s)	T2(s)	T3(s)	OPT	T1(s)	T2(s)	T3(s)
100	14735	212	152.42	16.4	15912	194	111.92	12.52	14046	147	156.33	14.46
120	13463	777	388.88	37.1	14198	1146	732.93	33.9	12940	773	681	39.36
140	12457	4352	2371	57.5	13298	2632	2180.55	55.74	12390	3245	2278	65.24
160	11969	*	5150	89.28	12693	6402	3330	87.32	11814	*	*	109.32
180	11682	*	8251	105.98	12049	*	6317	155.04	11647	*	5813	164.96
200	11532	*	*	237.46	11764	-	*	197.82	11096	-	*	349.82
220	11338	-	*	454.36	11461	-	-	317.76	11127	-	*	401.86
240	11401	-	-	652.4	11753	-	*	397.22	11690	-	*	413.56
249	11838	-	-	1301	12266	-	*	420.68	11690	-	*	731.44

Tabela 4: Resultados do  $MDF_{cutD}^r$  para  $n=500$  e  $d=3$

Vale mencionar que também usamos o CPLEX com as formulações  $MDF_{int}^r$  e  $MDF_{floD}^r$ ,  $MDF_{MTZ}^r$ , que possuem tamanho polinomial e podem ser aplicadas diretamente ao solver. Destacamos que nenhuma delas foi capaz de resolver qualquer das 147 instâncias ALM no tempo limite.

## 6. Novas Instâncias de Teste

Observamos que a maioria das instâncias ALM de Martins e Souza (2009), quando adaptadas para o MDF-MST, são resolvidas de forma ótima pela relaxação linear de  $MDF_{cutD}^r$  ou  $MDF_{SEC}^r$ , em tempo relativamente baixo. O mesmo acontece quando executamos o algoritmo exato da Seção 5 com o CPLEX. Por isso, resolvemos criar outras instâncias, potencialmente mais difíceis. Concretamente, procuramos gerar instâncias onde a solução da relaxação linear não é inteira e para as quais o solver CPLEX necessita de um maior tempo de computação para chegar à otimalidade.

Verificamos que três aspectos afetam a dificuldade do problema: (i)  $\beta := n - (\sum_{i \in C} d_i - c + 2)$ , (ii) a distribuição de  $d_i$ ,  $i \in C$ , e (iii) se a relação  $c_{ij} \leq c_{ik} + c_{kj}$  é satisfeita ou não, gerando instâncias euclidianas ou não euclidianas, respectivamente. Por Dias et. al.(2013) sabemos que  $\beta \geq 0$  para a instância ser viável. Usando tais observações, criamos 4 grupos de instâncias euclidianas e 4 grupos de não euclidianas da seguinte forma. Fixado um valor de  $c$ , determinamos o valor de  $n$  de modo a ter  $\delta = 0.4c$ , nos grupos 1 e 2, e  $\delta = 0$ , nos grupos 3 e 4. Em outras palavras, nesses dois últimos grupos, a quantidade de centrais é a máxima permitida para viabilidade, levando a instâncias mais difíceis. Os valores de  $c$  são 60, 100, 200, 300, 400, 500, 600, 700, 800 e 900. A diferenciação entre os grupos 1 e 2 e entre os grupos 3 e 4 deve-se à distribuição dos graus. Nos grupos 1 e 3, adotamos  $d_i = d = 3$ ,  $\forall i \in C$ . As instâncias euclidianas dos grupos 2 e 4 têm graus mínimos  $d_i \in \{2, 3, 4\}$ , distribuídos entre os centrais da seguinte forma: 25% com  $d_i = 2$ , 60% com  $d_i = 3$  e 15% com  $d_i = 4$ . Isto nos leva a  $\sum_{i \in C} d_i = 2.9c$ . Nos grupos 2 e 4 de instâncias não-euclidianas, usamos  $d_i \in \{3, 4, 5\}$ , assim distribuídos entre os vértices centrais: 60% com  $d_i = 3$ , 20% com  $d_i = 4$  e 20% com  $d_i = 5$ . Assim, temos  $\sum_{i \in C} d_i = 3.6c$ .

Os custos das instâncias euclidianas foram gerados pelo mesmo processo usado nas instâncias ALM de Martins e Souza(2009), por isso iremos manter a denominação ALM para es-

sas instâncias. Já para as instâncias não euclidianas (NEU), os custos serão definidos de forma aleatória, com valores uniformemente distribuídos dentro de um mesmo intervalo  $[1, 1000]$ .

Na Tabela 5 estão listados todos os conjuntos de instâncias, 80 no total.

Instâncias ALM						Instâncias NEU											
nome	$n$	$c$	nome	$n$	$c$	nome	$n$	$c$	nome	$n$	$c$	nome	$n$	$c$	nome	$n$	$c$
ALM60-1	146	60	ALM300-3	602	300	ALM600-4	1142	600	NEU60-1	146	60	NEU300-3	602	300	NEU600-4	1562	600
ALM60-2	140	60	ALM300-4	572	300	ALM700-1	1682	700	NEU60-2	182	60	NEU300-4	782	300	NEU700-1	1682	700
ALM60-3	122	60	ALM400-1	962	400	ALM700-2	1612	700	NEU60-3	122	60	NEU400-1	962	400	NEU700-2	2102	700
ALM60-4	116	60	ALM400-2	922	400	ALM700-3	1402	700	NEU60-4	158	60	NEU400-2	1202	400	NEU700-3	1402	700
ALM100-1	242	100	ALM400-3	802	400	ALM700-4	1332	700	NEU100-1	242	100	NEU400-3	802	400	NEU700-4	1822	700
ALM100-2	232	100	ALM400-4	762	400	ALM800-1	1922	800	NEU100-2	302	100	NEU400-4	1042	400	NEU800-1	1922	800
ALM100-3	202	100	ALM500-1	1202	500	ALM800-2	1842	800	NEU100-3	202	100	NEU500-1	1202	500	NEU800-2	2402	800
ALM100-4	192	100	ALM500-2	1152	500	ALM800-3	1602	800	NEU100-4	262	100	NEU500-2	1502	500	NEU800-3	1602	800
ALM200-1	482	200	ALM500-3	1002	500	ALM800-4	1522	800	NEU200-1	482	200	NEU500-3	1002	500	NEU800-4	2082	800
ALM200-2	462	200	ALM500-4	952	500	ALM900-1	2162	900	NEU200-2	602	200	NEU500-4	1302	500	NEU900-1	2162	900
ALM200-3	402	200	ALM600-1	1442	600	ALM900-2	2072	900	NEU200-3	402	200	NEU600-1	1442	600	NEU900-2	2702	900
ALM200-4	382	200	ALM600-2	1382	600	ALM900-3	1802	900	NEU200-4	522	200	NEU600-2	1802	600	NEU900-3	1802	900
ALM300-1	722	300	ALM600-3	1202	600	ALM900-4	1712	900	NEU300-1	722	300	NEU600-3	1202	600	NEU900-4	2342	900
ALM300-2	692	300							NEU300-2	902	300						

Tabela 5: Instâncias de teste das classes ALM e NEU

## 6.1. Avaliação Computacional das Relaxações Lineares

Também para essas novas instâncias, realizamos uma avaliação computacional das relaxações lineares, como foi feito na Seção 4. Na Tabela 6, encontra-se um resumo dos resultados. Como cada formulação resolveu uma quantidade diferente de instâncias no tempo limite, para realizarmos uma comparação condizente, essa tabela considera apenas as instâncias resolvidas por todas as formulações, ou seja, as 8 primeiras instâncias das classes ALM e NEU. Na Tabela 7 estão os valores individuais para as instâncias para as quais a relaxação linear de pelo menos uma das 5 formulações foi efetivamente resolvida. Onde se encontra “-” houve falta de memória; já “\*” marca os casos onde foi ultrapassado o tempo máximo permitido de 9000 segundos.

As formulações  $MDF_{SEC}$ ,  $MDF_{cutD}^r$  e  $MDF_{int}$  encontram limite significativamente melhor que aqueles encontrados pelas formulações  $MDF_{MTZ}^r$  e  $MDF_{int}$ . Quando comparamos os tempos computacionais, percebemos que  $MDF_{SEC}$  e  $MDF_{cutD}^r$  são comparáveis e bem mais efetivas que  $MDF_{int}$ . As formulações  $MDF_{MTZ}^r$  e  $MDF_{floD}^r$  são as mais rápidas, com ganho da primeira (embora tenha perdido nas 8 instâncias menores, conforme Tabela 6).

Formulações	Instâncias ALM			Instâncias NEU		
	Média GAP	Desvio Padrão GAP	Média Tempo (s)	Média GAP	Desvio Padrão GAP	Média Tempo (s)
CutsetD <sup>r</sup>	0,000168313	0,000336921	5,18	0,000254353	0,00048327	5,12
Subtour	0,000168313	0,000336921	5,43	0,000254353	0,00048327	4,93
Inter	0,000168313	0,000336921	318,63	0,000254353	0,00048327	239,52
MTZ <sup>r</sup>	0,090528201	0,046779017	0,61	0,00583772	0,005960462	0,76
FlowD <sup>r</sup>	0,080952317	0,008330155	4,64	0,070631203	0,02686164	4,64

Tabela 6: Média e desvio padrão dos GAP e média dos tempos para 8 instancias ALM e NEU

## 6.2. Resultados Exatos

Na Tabela 8, encontram-se os resultados do algoritmo descrito na Seção 5 para as instâncias onde a solução ótima foi encontrada.

O algoritmo de planos de corte foi capaz de resolver todas as instâncias das duas classes até 300 vértices centrais e uma instância ALM com  $c = 400$  do grupo 4. O mesmo não aconteceu para instâncias com  $c \geq 400$ . Algumas instâncias pararam por ultrapassar o tempo máximo de execução, e a grande maioria parou por falta de memória. Das 33 instâncias que foram resolvidas, em 12 a relaxação linear é inteira e em 21 não; e nessas 21 instâncias, o valor da relaxação linear é muito forte, bem próximo do valor da solução ótima. Como esperado, as novas instâncias geradas trouxeram maior dificuldade ao solver CPLEX, que não foi capaz de resolver a maioria delas.

Problema	CutsetD	T(s)	Subtour	T(s)	MTZ	T(s)	Inter	T(s)	FlowD <sup>r</sup>	T(s)	OPT
ALM60-1	6943	1,28	6943	0,9	6490,5	0,2	6943	94,74	6360,05	0,64	6943
ALM60-2	6772	1,02	6772	0,78	6361,23	0,22	6772	16,86	6233,77	0,78	6772
ALM60-3	6709	0,94	6709	0,92	5866,28	0,22	6709	64,92	6085,31	1,08	6709
ALM60-4	7040	1	7040	0,9	5946,84	0,22	7040	110,54	6531,29	1	7040
ALM100-1	8209,5	11,36	8209,5	6,26	7887	1	8209,5	329,74	7470,12	8,3	8217
ALM100-2	8065,5	9,76	8065,5	5,32	7810,5	0,96	8065,5	395,98	7399,84	6,98	8069
ALM100-3	8055	9,4	8055	11,54	7171,58	0,88	8055	813	7451,39	10,02	8055
ALM100-4	8139	6,68	8139	16,84	7035,06	1,18	8139	723,32	7567,69	8,34	8139
ALM200-1	11509	327,68	11509	141,96	11011	9,54	-	-	10339,52	154,32	11509
ALM200-2	11288,25	282,18	11288,25	172,48	10722,02	8,54	-	-	10137,38	149,78	11305
ALM200-3	11433	252,32	11433	596,6	10014,72	9,78	-	-	10536,3	208,44	11454
ALM200-4	11362,5	205,18	11362,5	729,12	9836,9	9,9	-	-	10583,05	220,2	11391
ALM300-1	13457	2074	13457	2312	12813,5	35,58	-	-	12168,97	798,88	13462
ALM300-2	13002,5	2294,3	13002,5	1739	12375,17	34,46	-	-	11726,12	870	13004
ALM300-3	12779	932	*	-	11568,05	35,04	-	-	11785,79	1277,4	12797
ALM300-4	12519	698,86	12519	5863	11437,17	36,76	-	-	11606,99	1255,2	12545
ALM400-1	-	-	15514	8941	-	-	-	-	14001,38	2573	-
ALM400-2	-	-	15107	6562	-	-	-	-	13601,81	2931,32	-
ALM400-3	-	-	*	-	-	-	-	-	14543,95	4288	-
ALM400-4	15239,75	3480	-	-	-	-	-	-	14381,49	4075	15255
ALM500-1	-	-	-	-	-	-	-	-	15488,25	8546	-
ALM500-2	-	-	-	-	-	-	-	-	15397,34	6208	-
ALM500-3	-	-	-	-	-	-	-	-	*	-	-
NEU60-1	3032	1,26	3032	1,04	3032	0,24	3032	38,7	2769,8	0,7	3032
NEU60-2	3872	0,66	3872	0,14	3870	0,24	3872	11,64	3672,71	0,94	3872
NEU60-3	2970,5	0,92	2970,5	0,38	2941,41	0,28	2970,5	163,56	2721,22	1,16	2972
NEU60-4	4031	0,86	4031	0,4	3970,52	0,3	4031	37,64	3841,43	1,16	4031
NEU100-1	3079	13,8	3079	12,64	3079	0,88	3079	523,6	2808,23	7,6	3079
NEU100-2	3858	10,74	3858	12,5	3852	1,16	3858	260,62	3688,15	7,82	3858
NEU100-3	3083,5	8,08	3083,5	4,44	3056	1,64	3083,5	543,82	2724,11	8,02	3084
NEU100-4	3650	4,68	3650	7,9	3617,52	1,34	3650	336,62	3483,1	9,78	3655
NEU200-1	3326	527,5	3326	324,44	3326	8,96	-	-	3078,96	123,02	3326
NEU200-2	4280	446,84	4280	226,96	4266	14,46	-	-	4014,17	134,64	4280
NEU200-3	3386,5	226,32	3386,5	199,34	3352,04	19,26	-	-	3104,55	202,46	3388
NEU200-4	4221,56	328	4221,56	358,78	4176,04	21,84	-	-	3987,71	244,96	4223
NEU300-1	3462	5359	*	-	3459	44,44	-	-	3196,32	794,92	3462
NEU300-2	4382	5331	*	-	4379	70,42	-	-	4130,89	917	4382
NEU300-3	3323,01	2365,56	3323,01	2999,54	3295,68	92,8	-	-	3002,18	1219,94	3326
NEU300-4	4000,33	2579	*	-	3960,22	121,96	-	-	3724,82	1422,58	4001
NEU400-1	-	-	-	-	-	-	-	-	3293,72	2494	-
NEU400-2	-	-	-	-	-	-	-	-	3968,51	2946,14	-
NEU400-3	-	-	-	-	-	-	-	-	3184,07	4938	-
NEU400-4	-	-	-	-	-	-	-	-	3924,04	5094	-
NEU500-1	-	-	-	-	-	-	-	-	3245,95	6990	-
NEU500-2	-	-	-	-	-	-	-	-	4130,69	6575	-
NEU500-3	-	-	-	-	-	-	-	-	*	-	-

Tabela 7: Comparação dos limites de Programação Linear para instâncias de Teste ALM e NEU

Instâncias ALM						Instâncias NEU					
nome	OPT	T(s)	nome	OPT	T(s)	nome	OPT	T(s)	nome	OPT	T(s)
ALM60-1	6943	1,28	ALM200-1	11509	328,68	NEU60-1	3032	1,26	NEU200-1	3326	527,5
ALM60-2	6772	1,02	ALM200-2	11305	332,64	NEU60-2	3872	0,66	NEU200-2	4280	446,84
ALM60-3	6709	0,94	ALM200-3	11454	282,82	NEU60-3	2972	1,02	NEU200-3	3388	236,7
ALM60-4	7040	1	ALM200-4	11391	648,4	NEU60-4	4031	0,86	NEU200-4	4223	360,44
ALM100-1	8217	11,96	ALM300-1	13462	2092	NEU100-1	3079	13,8	NEU300-1	3462	5361
ALM100-2	8069	10,24	ALM300-2	13004	2312,2	NEU100-2	3858	10,74	NEU300-2	4382	5351
ALM100-3	8055	9,4	ALM300-3	12797	1039	NEU100-3	3084	8,72	NEU300-3	3326	2686
ALM100-4	8139	6,68	ALM300-4	12545	998,22	NEU100-4	3655	6,38	NEU300-4	4001	2891
			ALM400-4	15255	3621						

Tabela 8: Resultado do algoritmo exato para as novas instâncias das classes ALM e NEU

## 7. Conclusão e Perspectivas

Neste trabalho apresentamos cinco formulações para o MDF–MST, baseadas em formulações para árvores geradoras. Realizamos uma comparação teórica dos limites de relaxação linear destas formulações para o MDF–MST. Também realizamos um estudo computacional destes limites para instâncias da literatura e novas instâncias introduzidas neste trabalho. Mostramos serem tais limites bastante apertados para quase todas as instâncias consideradas.

Implementamos uma nova versão do algoritmo de planos de corte, usando o CPLEX, apresentado por Dias et al. (2013). Essa nova versão se diferencia na política de adição de restrições violadas. Optamos por adicionar todas as restrições violadas encontradas, evitando que restrições iguais sejam adicionadas. Podemos observar pelos resultados experimentais que essa nova versão é computacionalmente mais eficiente que as duas versões apresentadas anteriormente em Dias et al. (2013).

Temos resultados preliminares, utilizando a relaxação lagrangeana com a formulação  $MST_{cutD}^r$ . O algoritmo lagrangeano encontra em geral bons limites superiores e inferiores em tempo competitivo com aqueles aqui apresentados. Estamos trabalhando no sentido de reduzir o gap encontrado por esse algoritmo lagrangeano e torná-lo mais rápido.

### Referências

- [1] **Achuthan, N.; Caccetta, L.; Caccetta, P.; Geelen, J.** *Computational methods for the diameter restricted minimum weight spanning tree problem.* Australasian Journal of Combinatorics, v. 10, p. 51-71, 1994.
- [2] **Almeida, A. M.; Martins, P.; Souza, M.C.** *Min-Degree Constrained Minimum Spanning Tree Problem: Complexity, properties and formulations.* International Transactions in Operational Research, v. 19, p. 323-352, 2012.
- [3] **Dantzig, G. B.; Fulkerson, D. R.; Johnson, S. M.** *Solution of a large scale traveling salesman problem.* Operations Research, v. 2, p. 393-410, 1954.
- [4] **Dias, F. C. S.; Campêlo, M.; Andrade, R. C.; de Souza, C. P.** *Problema de Árvore Geradora Mínima com Restrição de Grau Mínimo e Centrais Fixas.* In: XLV Simpósio Brasileiro de Pesquisa Operacional - SBPO, 2013, Natal. Anais do Simpósio, 2013. p. 2438-2449.
- [5] **Gouveia, L.** *Using the Miller-Tucker-Zemlin constraints to formulate a minimal spanning tree problem with hop constraints.* Computers & Operations Research, v. 22(9), p. 959–970, 1995.
- [6] **Gouveia, L.; Telhada, J.** *Reformulation by Intersection Method on the MST Problem with Lower Bound on the Number of Leaves.* Lecture Notes in Computer Science, v. 6701, p. 83-91, 2011.
- [7] **Magnanti, T. L.; Wolsey, L. A.** *Optimal Trees,* Handbooks in Operations Research and Management Science - Elsevier, v. 7, c. 9, p. 503-616, 1995.
- [8] **Martin, R. K.** *Using separation algorithms to generate mixed integer model reformulations.* Operations Research Letters, v. 10, p. 119–128, 1991.
- [9] **Martins, P.; Souza, M. C.** *VNS and second order heuristics for the min-degree constrained minimum spanning tree problem.* Computers and Operations Research, v. 36, p. 2669–2982, 2009.
- [10] **Martinez, L.C.; Cunha, A.** *A Parallel Lagrangian Relaxation Algorithm for the Min-Degree Minimum Spanning Tree Problem.* In: 2nd International Symposium on Combinatorial Optimization, ISCO 2012, Atenas. Lecture Notes in Computer Science - Combinatorial Optimization. Heidelberg: Springer-Verlag, v. 7422. p. 237-248, 2012.
- [11] **Miller, C; Tucker, A; Zemlin, R.** *Integer programming formulation of traveling salesman problems.* Journal of ACM, v. 7, p. 326-9, 1960.
- [12] **Narula, S. C.; Ho, C. A.** *Degree-constrained minimum spanning tree.* Computers and Operations Research, v. 7, p. 239-249, 1980.