

Problema de Árvore Geradora Mínima com Restrição de Grau Mínimo e Centrais Fixos¹

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 Sousa

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

RESUMO

Apresentamos uma variação do Problema de Árvore Geradora Mínima com Restrição de Grau Mínimo, na qual os vértices folhas são definidos a priori. Derivamos algumas propriedades e condições de viabilidade para o problema. Provamos que é NP-Difícil no caso geral e também sob várias restrições de parâmetros de entrada, como o grafo e o grau mínimo dos vértices. Apresentamos uma formulação de programação inteira baseada em restrições de eliminação de cortes. Reportamos experiências computacionais para várias instâncias de teste, mostrando que a formulação é bastante eficaz. Praticamente todas as instâncias foram resolvidas à otimalidade, muitas delas pela relaxação linear.

PALAVRAS CHAVE. Árvore geradora mínima com restrição de grau, NP-difícil, Formulação matemática.

ABSTRACT

We present a variation of the Min-Degree Constrained Minimum Spanning Tree Problem, where the leaves are fixed a priori. We derive some properties and feasibility conditions for the problem. We prove that it is NP-hard in the general case and under several restrictions on some parameters such as the input graph and the minimum degree of the vertices. We present an ILP formulation based on cut elimination constraints. We report computational experiments for benchmark instances showing that the fomulation is very effective. Almost all instances were solved to optimality, many of them by the linear relaxation.

KEYWORDS. Degree-constrained minimum spanning tree, NP-hard, ILP formulation.

1 Introdução

O problema de Árvore Geradora Mínima (*Minimum Spanning Tree*, MST) consiste em encontrar, em um grafo ponderado em arestas, uma árvore geradora de custo mínimo. Na literatura, tem-se estudado diversas variações do MST, onde se impõem restrições adicionais ao problema. Podemos citar o *Capacitated Minimum Spanning Tree* (Esau; Williams, 1966), *Maximum Leaf Spanning Tree* (Garey; Johnson, 1979), *Degree Constrained Minimum Spanning Tree* (Narula; Ho, 1980), *Bounded Diameter Minimum Spanning Tree* (Achuthan

¹Financiado parcialmente por CNPq (Proc. 307627/2010-1, 480608/2011-3, 132998/2011-4), e Projetos FUNCAP/PRONEM and CAPES/STIC-AmSud.

et al., 1994), *Generalized Minimum Spanning Tree* (Myung et al., 1995), *Hop-Constrained Minimum Spanning Tree* (Gouveia, 1996), *Delay-Constrained Minimum Spanning Tree* (Sallama et al., 1997), como exemplos de problemas conhecidos relacionados ao MST.

Entre essas variações do MST, enfatizamos aquelas com restrições sobre os graus dos vértices na árvore geradora. Seja $G = (V, E)$ um grafo conexo e não direcionado com $|V| = n$ e $|E| = m$. Associe a cada aresta $e \in E$ um custo $c_e \geq 0$ e a cada vértice $v \in V$ um inteiro positivo $d(v)$, relacionado ao grau de v na árvore.

O problema de *Árvore Geradora Mínima com Restrição de Grau* (*Degree-Constrained Minimum Spanning Tree Problem*, dc-MST) consiste em encontrar uma árvore geradora mínima de G onde o grau de cada vértice v , na árvore, é no máximo $d(v)$. Este é um problema NP-difícil (Narula; Ho, 1980). Já no problema de *Árvore Geradora Mínima com Restrição de Grau Mínimo* (*Min-Degree Constrained Minimum Spanning Tree*, md-MST) deseja-se encontrar uma árvore geradora mínima de G , tal que cada vértice v tem grau pelo menos $d(v)$ ou é uma folha (grau 1) na árvore. Este problema foi introduzido por Almeida, Martins e Souza (2006), que apresentaram sua complexidade, propriedades, formulações e um algoritmo branch-and-bound. Outras formulações e resultados computacionais para md-MST podem ser vistos em (Martinez; Cunha, 2012; Martinez; Cunha, 2013).

Na literatura, normalmente considera-se $d(v) = d, \forall v \in V$, e chama-se os vértices com grau mínimo d de centrais e as folhas, de terminais. Observe que no md-MST os vértices centrais e terminais, ou mesmo suas quantidades, são arbitrários. Neste trabalho, descrevemos uma variação do problema, onde os vértices centrais e terminais são fixados a priori. Vamos chamá-lo de *Problema de Árvore Geradora Mínima com Restrição de Grau Mínimo e Centrais Fixos* (mdf-MST). A Figura 1 ilustra uma instância do problema para $d = 2$. Note que, em toda árvore viável T , o subgrafo de T induzido pelos centrais é uma subárvore. Em uma solução ótima, porém, essa subárvore pode não ser geradora mínima para o subgrafo induzido pelos centrais.

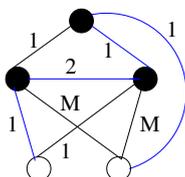


Figura 1: Instância de mdf-MST com $d = 2$. Vértices centrais em preto. Solução ótima em azul para $M > 2$.

Aplicações práticas para o mdf-MST aparecem em projetos de redes para sistemas VLSI, estradas, redes de energia e computadores, quando se deseja uma rede mínima de conexão com nós especiais onde a informação ou instalações são centralizadas. Esses nós, ditos centrais (que podem ser centros de distribuição ou dispositivos de comunicação centralizados) e os nós terminais (que podem ser clientes ou dispositivos periféricos) devem se conectar, de forma que cada nó central deve se ligar a um número mínimo de outros nós centrais ou terminais, e estes últimos devem se conectar a um único nó central.

A versão não-ponderada dos problemas dc-MST, md-MST e mdf-MST será denotada por dc-ST, md-ST e mdf-ST, respectivamente (retira-se o “M” da sigla). Esses são problemas de decisão, onde se deseja verificar se existe ou não uma árvore com as correspondentes restrições de grau. Note, porém, que eles são equivalentes às versões de otimização onde os custos são unitários, pois neste caso o custo de qualquer solução ótima é $n - 1$.

Aqui, estudamos os problemas mdf-ST e mdf-MST. Provamos sua complexidade computacional, apresentar propriedades e formulações e reportar experiências computacionais com algoritmos de solução. Também consideramos uma versão particular do problema

onde, no grafo original, cada terminal se liga a todos os centrais. Obtemos assim os problemas mdf-ST-TC e mdf-MST-TC, correspondendo ao caso não ponderado e ponderado respectivamente. Mostraremos que mesmo essas versões são NP-Difíceis.

2 Condições de Viabilidade

Adotamos a seguinte notação associada ao grafo $G = (V, E)$. Particionamos $V = C \cup T$, com $|C| = c$, $|T| = t$, $c + t = n$, onde C e T descrevem os conjuntos de centrais e terminais, respectivamente. Denotamos por $E(C)$ o conjunto de arestas entre os vértices centrais, e por $E(C, T)$ as arestas entre centrais e terminais. Por simplicidade, consideramos que não há arestas em G ligando dois terminais, já que elas nunca farão parte de uma solução. O subgrafo induzido pelos vértices centrais será denotado por $G(C)$. Consideramos que $d(v) = d, \forall v \in C$, e denotar por d_v o grau do vértice v numa solução viável qualquer.

Para que os problemas em estudo possam ser viáveis, supomos que o grau, em G , de cada vértice de C é maior ou igual a d e que $G(C)$ é conexo. Também supomos que $d \geq 2$, pois o problema se resume ao MST se $d = 1$.

Almeida, Martins e Souza (2006) apresentaram propriedades para o md-MST que nos fornecem limites inferiores e superiores para a quantidade de vértices centrais e terminais. Tal resultado se transfere para o mdf-MST como segue.

Proposição 1 *Se mdf-MST é viável então $n - \lfloor (n - 2)/(d - 1) \rfloor \leq t \leq n - 1$ e $1 \leq c \leq \lfloor (n - 2)/(d - 1) \rfloor$.*

Uma condição similar, da qual se deriva a Proposição 1, para a viabilidade do mdf-MST é dada abaixo, relacionando c , t e d .

Proposição 2 *Se mdf-MST é viável então $t \geq c(d - 2) + 2$.*

Prova. Em qualquer solução viável do mdf-ST, temos que $\sum_{v \in V} d_v = 2(c + t - 1)$ e $\sum_{v \in V} d_v \geq cd + t$. Logo, $2(c + t - 1) \geq cd + t$, ou seja, $t \geq c(d - 2) + 2$. ■

Note que, ao usarmos $c + t = n$ na Proposição 2, chegamos ao resultado da Proposição 1. Particularmente, para o mdf-MST ser viável temos como valor mínimo e máximo para t , $t_{min} = c(d - 2) + 2$ e $t_{max} = n - 1$. Similarmente, o valor mínimo e máximo para c é $c_{min} = 1$ e $c_{max} = \lfloor (t - 2)/(d - 2) \rfloor$ (se $d > 2$) ou $c_{max} = n - 2$ (se $d = 2$).

Corolário 3 *Suponha que mdf-MST é viável. Então $d \leq \lfloor \frac{n-2}{c} \rfloor + 1$. Em particular, se $c \geq 2$, então $d \leq \lfloor \frac{n}{2} \rfloor$*

Prova. Substituindo $t = n - c$ em $t \geq c(d - 2) + 2$ chegamos ao primeiro resultado. Se $c \geq 2$, então $d \leq \lfloor \frac{n-2}{2} \rfloor + 1 = \lfloor \frac{n}{2} \rfloor$. ■

Os resultados acima estabelecem condições necessárias para a viabilidade de mdf-MST, todas elas derivadas de $t \geq c(d - 2) + 2$. Esta condição, porém, não é suficiente. De fato, se $d = 2$, a condição torna-se $t \geq 2$. Porém, se $G(C)$ não possui um caminho Hamiltoniano, o problema é inviável para $t = 2$. Esse é o caso da Figura 2, onde descrevemos $G(C)$. Considerando $d = 2$, precisamos de pelo menos 4 terminais para completar o grau dos vértices em C , embora $t_{min} = 2$.

Temos que $t_{min} = c(d - 2) + 2 = cd - (2c - 2) = cd - 2(c - 1)$. Analisando $cd - 2(c - 1)$, chegamos à conclusão que $2(c - 1)$ é a contribuição que as arestas conectando vértices de C fornecem ao grau dos vértices de C em qualquer solução, já que para qualquer solução, digamos H , de mdf-ST, $H(C)$ também deve ser uma árvore. Com isso, t_{min} seria o valor mínimo necessário para gerar uma solução viável H para o problema se tivéssemos a árvore

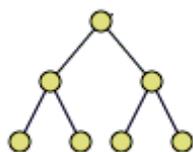


Figura 2: Exemplo de subgrafo $G(C)$

$H(C)$ com grau máximo menor ou igual a d , e os vértices com grau menor do que d ligados a uma quantidade suficiente de terminais para completar seu grau mínimo. Como podemos ter vértice em $H(C)$ com grau maior que d , o número de terminais deve ser suficiente para compensar a soma dos excessos em relação a d na árvore $H(C)$, ou seja, $t \geq t_{min} + \sum_{v \in C} \{\max\{0, d_v - d\}\}$. Mas além de termos terminais em número suficiente, eles precisariam estar corretamente ligados, em G , aos vértices com grau menor que d em $H(C)$. No exemplo da Figura 2, quatro terminais seriam suficientes apenas se existisse em G um emparelhamento entre eles e as folhas de $G(C)$. Esta observação nos leva à seguinte caracterização da viabilidade de mdf-ST.

Proposição 4 *O problema mdf-MST é viável para a instância (G, d) se, e somente se, existe uma árvore geradora $A = (C, E_A)$ de $G(C)$ tal que o problema mdf-MST é viável para a instância (G', d) , onde $G' = (C \cup T, E_A \cup E(C, T))$.*

Convém então investigar o problema quando $G(C)$ é uma árvore. Neste caso, o mdf-ST se resume a verificar se é possível ou não ligar os terminais aos vértices de $G(C)$ com grau menor que d , de forma a satisfazer as restrições de grau mínimo.

Proposição 5 *Se o grafo induzido $G(C)$ é uma árvore então o mdf-ST é polinomial.*

Prova. Nossa prova se resume a mostrar como podemos verificar, em tempo polinomial, se existe uma solução viável para o mdf-ST a partir de $G(C)$. Reduzimos nosso problema a um problema de Fluxo Máximo. Seja o grafo $G = (C \cup T, E)$ com $G(C)$ uma árvore. Criamos um grafo direcionado $D = (\bar{V}, A)$ com dois vértices especiais $s, t \in \bar{V}$, uma instância do problema de Fluxo Máximo. Seja \bar{C} os vértices em C com grau menor que d em $G(C)$. Então fazemos $\bar{V} = \bar{C} \cup T \cup \{s, t\}$, onde s é a origem e t é o destino. Para cada aresta $\{i, j\} \in E$, com $i \in T$ e $j \in \bar{C}$, criamos o arco (i, j) com capacidade 1 em A . Criamos os arcos $(s, i), \forall i \in T$, com capacidade 1. Também criamos os arcos $(j, t), \forall j \in \bar{C}$, com capacidade $(d - d_j) \geq 1$ onde d_j é o grau do vértice j na árvore $G(C)$. Um fluxo máximo em D define uma alocação dos vértices em T aos vértices em \bar{C} . Sendo assim, temos que mdf-ST é viável se, e somente se, o fluxo máximo em D , entre s e t , é $\sum_{j \in \bar{C}} (d - d_j)$. ■

Proposição 6 *Se o grafo induzido $G(C)$ é uma árvore, então mdf-MST é polinomial.*

Prova. Verificar a viabilidade de mdf-MST é polinomial pela Proposição 5. Se viável, resolver mdf-MST equivale a resolver um problema de fluxo de custo mínimo. Criamos um grafo direcionado $D = (\bar{V}, A)$ com dois vértices especiais $s, t \in \bar{V}$, uma instância do problema de Fluxo de Custo Mínimo. Então fazemos $\bar{V} = C \cup T \cup \{s, t\}$, onde s é a origem e t é o destino. Para cada aresta $\{i, j\} \in E$, com $i \in T$ e $j \in C$, criamos o arco (i, j) em A , com capacidade 1 e custo igual ao custo da aresta $\{i, j\}$ em G . Criamos os arcos $(s, i), \forall i \in T$, com capacidade 1 e custo 0. Também criamos os arcos $(j, t), \forall j \in C$, com capacidade ∞ e custo 0. A oferta da origem s é $b_s = |T|$. A demanda de um vértice $u \in T$ é $b_u = 0$ e de um vértice $v \in C$ é $b_v = -\max\{0, d - d_v\}$, onde d_v é o grau do vértice v na árvore $G(C)$. Já a demanda do sumidouro t será dada por $b_t = -b_s - \sum_{v \in C} b_v$. É fácil verificar que uma solução de mdf-MST é dada por um fluxo de custo mínimo em D . ■

3 Complexidade de mdf-ST

Fazemos uma análise da complexidade do mdf-ST, provando que se trata de um problema NP -Difícil. Para isso mostraremos que o problema de decisão relacionado ao mdf-MST é NP-Completo. Reduzimos o Problema de Caminho Hamiltoniano para o mdf-ST. Um caminho Hamiltoniano em um grafo é um caminho que passa por cada vértice uma única vez. O problema de decidir se um grafo tem ou não um caminho Hamiltoniano entre dois vértices (HAM-CAM) é NP-Completo (Garey; Johnson, 1979).

Primeiramente, consideramos o problema mdf-ST restrito a um d fixo e denotá-lo por mdf-ST_d . Como consequência obtemos a complexidade de mdf-ST.

Teorema 7 *O Problema de Decisão da Árvore Geradora com Restrição de Grau Mínimo e Centrais Fixos com d fixo (mdf-ST_d) é NP-Completo.*

Prova. Primeiramente, mostramos que mdf-ST_d pertence a NP. Dado um grafo $G = (V, E)$, nosso certificado é o conjunto de arestas que formam a árvore geradora. O algoritmo de verificação confere se todos os vértices de C tem grau no mínimo d , se todos os vértices de T são folhas e se o grafo é realmente uma árvore geradora. Essa verificação pode ser feita em tempo polinomial.

Agora provamos que $\text{HAM-CAM} \leq_p \text{mdf-ST}_d$, o que mostra que mdf-ST_d é NP-Completo. Considere um grafo $G = (V, E)$ e os vértices (s, t) de V , ou seja, uma instância do problema HAM-CAM. Nosso algoritmo de redução irá criar um grafo $\bar{G} = (C \cup T, \bar{E})$ que será a instância para o mdf-ST_d . Começamos fazendo todos os vértices de V como vértices centrais em \bar{G} , ou seja, $V = C$, e induzindo as arestas de E em \bar{E} . Para cada vértice central $u \in C$ diferente de s e t , criamos $d - 2$ novos vértices que se conectam apenas a u . Criamos ainda $d - 1$ vértices que se conectam apenas a s , e $d - 1$ vértices que se conectam apenas a t . Todos os vértices criados formam o conjunto T , e as arestas criadas pertencem a \bar{E} .

Devemos mostrar que essa transformação é uma redução, ou seja, G tem um caminho Hamiltoniano entre s e t se, e somente se, o grafo \bar{G} tem uma árvore geradora onde o grau de todos os vértices em C seja no mínimo d e todos os vértices em T sejam folhas. Suponha que o grafo G tenha um caminho Hamiltoniano entre s e t . Esse caminho em \bar{G} juntamente com vértices e arestas criados formam uma árvore geradora para \bar{G} onde o grau de todos os vértices em C é d e todas as folhas estão em T . Suponha agora que o grafo \bar{G} tenha uma árvore geradora H onde o grau de todos os vértices em C seja no mínimo k e todos os vértices em T sejam folhas. Como cada vértice de T tem grau 1 em \bar{G} , então as arestas de T para C estão na árvore H . Isso obriga que os vértices s e t se conectem a pelo menos um vértice de $C \setminus \{s, t\}$, e os outros vértices centrais devem se conectar a pelo menos dois outros vértices de C . A única forma de satisfazer estas restrições e mantendo $H(C)$ uma árvore é construindo um caminho Hamiltoniano entre s e t . Concluimos que o grafo induzido $H(C)$ é um caminho Hamiltoniano entre s e t . ■

Corolário 8 *O Problema de Decisão da Árvore Geradora com Restrição de Grau Mínima e Centrais Fixo (mdf-ST) é NP-Completo.*

Prova. Considere o mdf-ST geral onde o valor de d é uma entrada do problema. Note que d é no máximo o número de vértices. Usando o Teorema 7, para qualquer valor de entrada d , o mdf-ST_d é NP-Completo. Com isso mdf-ST é NP-Completo. ■

Como consequência do Corolário 8, derivamos a complexidade de mdf-MST.

Corolário 9 *O Problema da Árvore Geradora Mínima com Restrição de Grau Mínimo e Centrais Fixos (mdf-MST) é NP-Difícil.*

4 Propriedades do mdf-ST-TC

Chamamos de mdf-ST-TC o mdf-ST quando cada vértice em T se liga a todos os vértices em C no grafo G . Nesse caso, podemos concluir que, se temos terminais suficientes, resolver o problema consiste em encontrar uma árvore geradora de $G(C)$ e, para cada vértice de C com grau menor que d na árvore geradora, completamos o grau ligando-o a vértices de T . É fácil ver que a maior quantidade de terminais é necessária quando $G(C)$ for um grafo estrela. Para esse caso, precisamos ter $t \geq (d-1)(c-1)$. Definindo então $\bar{t} = (d-1)(c-1)$, chegamos à seguinte conclusão.

Proposição 10 *Se $t < t_{min}$ ou $t \geq \bar{t}$, então mdf-ST-TC é polinomial.*

Pelo resultado acima, deduzimos que o problema é polinomial se $t_{min} \leq \bar{t}$, pois uma das duas condições é verificada. Comparando o valor de $\bar{t} = (d-1)(c-1)$ e $t_{min} = c(d-2) + 2$, temos

$$\begin{aligned} t_{min} &= c(d-2) + 2 = (c-1+1)(d-1-1) + 2 = (c-1)(d-1) + d + 1 - c \\ t_{min} &= \bar{t} + d + 1 - c \end{aligned}$$

Logo, $t_{min} \leq \bar{t}$ se, e somente se, $c \geq d + 1$. Pela proposição acima, chegamos a

Corolário 11 *Para $G = (C \cup T, E)$ e $c \geq d + 1$ o problema mdf-ST-TC é polinomial.*

O seguinte resultado é uma especialização da Proposição 4 para o mdf-ST-TC.

Proposição 12 *Existe uma solução do mdf-ST-TC se, e somente se, existe árvore geradora de $G(C)$ onde, $\forall P \subseteq C$, $\sum_{v \in P} d_v \leq t - t_{min} + pd$, sendo $p = |P|$.*

Prova. (\implies) Prova por absurdo. Suponha que toda árvore geradora de $G(C)$ possui P , subconjunto de C , tal que $\sum_{v \in P} d_v > t - t_{min} + pd$. Então,

$$\sum_{v \in C \setminus P} d_v = 2(c-1) - \sum_{v \in P} d_v < 2(c-1) - (t - t_{min} + pd) = (c-p)d - t$$

ou seja,

$$\sum_{v \in C \setminus P} d_v + t < (c-p)d$$

Logo, mesmo usando os t terminais para aumentar o grau dos vértices de $C \setminus P$, não se poderia atingir o grau total mínimo deles, no caso, $d(c-p)$. Portanto, não existe árvore geradora de $G(C)$ com grau mínimo d .

(\impliedby) Suponha que exista árvore geradora de $G(C)$ onde, $\forall P \subseteq C$, $\sum_{v \in P} d_v \leq t - t_{min} + pd$. Particularmente, esta desigualdade vale quando tomamos P como os vértices com grau maior que d na árvore geradora. Defina, $\forall v \in C \setminus P$, $\alpha_v = d - d_v$. Temos que,

$$\sum_{v \in C \setminus P} (d - \alpha_v) = \sum_{v \in C \setminus P} d_v = 2(c-1) - \sum_{v \in P} d_v \geq 2(c-1) + t_{min} - t - pd = (c-p)d - t$$

Logo, $\sum_{v \in C \setminus P} \alpha_v \leq t$. Isto mostra que os t terminais disponíveis são suficientes para completar o grau dos vértices em $C \setminus P$ e, assim, gerar uma árvore com grau mínimo d . ■

Corolário 13 *Se $t \geq t_{min}$ e existe uma árvore geradora de $G(C)$ de grau máximo d , então existe uma solução para o mdf-ST-TC, e ela é dada pela árvore geradora de grau máximo d de $G(C)$, acrescentando aos vértices de grau menor que d arestas para os vértices de T .*

Corolário 14 *Suponha que $t = t_{min}$. Existe solução para mdf-ST-TC se, e somente se, existe árvore geradora de $G(C)$ de grau máximo menor ou igual a d .*

O corolário acima mostra que mdf-ST-TC equivale, em alguns casos, ao problema dc-ST, que é NP-Completo.

Proposição 15 *O problema mdf-ST-TC é NP-Completo.*

Pelo Corolário 13, se $t \geq t_{min}$ e $G(C)$ tem um Caminho Hamiltoniano, então mdf-ST-TC é viável. Utilizando o Teorema de Dirac, que fornece uma condição suficiente para que um grafo seja Hamiltoniano, chegamos a uma condição que garante a viabilidade e a resolução em tempo polinomial do problema.

Proposição 16 *Se o grafo $G(C)$ possui grau mínimo maior ou igual a $\lceil n/2 \rceil$, então $G(C)$ é Hamiltoniano e mdf-ST-TC é polinomial.*

Prova. Aplicando o Teorema de Dirac, temos que se $G(C)$ possui grau mínimo maior ou igual a $\lceil n/2 \rceil$ então $G(C)$ é Hamiltoniano e um caminho hamiltoniano pode ser obtido em tempo polinomial. Para atingir o grau mínimo d , a partir desse caminho, precisamos de apenas $(c-2)(d-2) + 2(d-1) = c(d-2) + 2 = t_{min}$ terminais. Logo, mdf-ST-TC é polinomial. ■

Outra condição suficiente para que o problema seja polinomial, mais fraca do que aquelas usadas nas proposições 10 e 16, é dada a seguir.

Proposição 17 *Seja \mathcal{F} o conjunto de árvores geradoras de $G(C)$. Seja $\Delta^* = \min\{\Delta(A) : A \in \mathcal{F}\}$. Se $d \geq \Delta^* + 1$, então mdf-ST-TC é polinomial.*

Prova. Uma árvore geradora R de $G(C)$ com grau máximo $\Delta(R) \leq \Delta^* + 1$ pode ser encontrada em tempo polinomial (Furer; Raghavachari, 1994). Se $t < t_{min}$, o problema é inviável. Do contrário, pelo Corolário 13, uma solução é obtida facilmente a partir de R . ■

5 Formulação Matemática

Como para muitos problemas relacionados ao MST, uma formulação matemática para o mdf-MST pode ser obtida por qualquer conjunto de restrições que definam uma árvore geradora e pelas restrições particulares do problema. Em (Magnanti; Wolsey, 1994) encontramos boa parte das formulações para árvore geradora. Podemos obter uma formulação para o mdf-MST utilizando, por exemplo, as restrições de cortes, restrições de eliminação de ciclos (Dantzig et al., 1954), restrições de Miller et al. (1960), dentre outras. Um detalhe importante é que, para o mdf-MST, podemos aplicar as restrições de árvore geradora apenas no subgrafo induzido $G(C)$.

Aqui, consideramos a versão orientada da formulação definida com restrições de cortes. Para isso, se faz necessário transformar o grafo não-direcionado $G = (V, E)$ em um grafo direcionado $D = (V, A)$. Substituímos cada aresta $e = \{i, j\} \in E$ por arcos (i, j) e (j, i) em A com custo igual c_e em ambas as direções. Como trabalhamos com o grafo direcionado, a árvore geradora resultante será direcionada e, com isso, devemos definir um vértice raiz $r \in C$. Podemos então desconsiderar os arcos de T para C .

Seja y_{ij} a variável binária que será igual a 1 se o arco (i, j) estiver na solução e 0 caso contrário. Para $S, S' \subset V$, sejam $A(S)$ e $\delta^+(S, S')$ o subconjunto de arcos de D com ambas as extremidades em S e com origem em S e destino em S' , respectivamente. Por simplicidade, usaremos $\delta^+(i) = \delta^+(\{i\}, V \setminus \{i\})$. Assim obtemos:

$$(mdf-MST_{cutsetD}) \quad \min \sum_{(i,j) \in A} c_{ij} y_{ij} \quad (1)$$

$$\text{sujeito a:} \quad \sum_{(i,j) \in A(C)} y_{ij} = c - 1 \quad (2)$$

$$\sum_{(i,j) \in \delta^+(S, C \setminus S)} y_{ij} \geq 1, \forall S \subset C \text{ e } r \in S \quad (3)$$

$$y_{ij} \in \{0, 1\}, \forall (i, j) \in A \quad (4)$$

$$\sum_{(r,j) \in \delta^+(r)} y_{rj} \geq d \quad (5)$$

$$\sum_{(i,j) \in \delta^+(i)} y_{ij} \geq d - 1, \forall i \in C \setminus \{r\} \quad (6)$$

$$\sum_{(i,t) \in \delta^+(C, T)} y_{it} = 1, \forall t \in T \quad (7)$$

Note que as restrições (2)-(4) induzem uma arborescência em $G(C)$. Já as restrições (6)-(7) asseguram o grau de cada vértice, observando se ele é a raiz, um central diferente da raiz ou um terminal. Em substituição a (2)-(4), poderíamos usar restrições de eliminação de ciclos ou de Miller et al. (1960), gerando assim duas outras formulações, $(mdf-MST)_{SEC}$ e $(mdf-MST)_{MTZ}$ respectivamente.

6 Método de solução

Utilizando o *solver* CPLEX, implementamos um algoritmo para o mdf-MST, baseado na formulação (1)-(7). O algoritmo começa resolvendo a relaxação linear. Caso a solução não seja inteira, passamos a resolver a formulação com todas as variáveis inteiras.

Note que a quantidade de restrições em (3) é exponencial, mas elas podem ser separadas em tempo polinomial, resolvendo problemas de corte mínimo no grafo (Magnanti; Wolsey, 1994). Assim, implementamos um procedimento de separação que determina a restrição (3) mais violada ou verifica que todas estão satisfeitas.

Começamos o modelo relaxado sem as restrições (3) e as acrescentamos iterativamente, uma por vez, usando o procedimento de separação. Após encontrarmos a solução relaxada, se necessário, passamos a resolver o modelo com todas as variáveis inteiras. Mantemos as restrições do tipo (3) já adicionadas à relaxação linear e reaplicamos o algoritmo de separação como antes.

Com alguns testes computacionais, verificamos que o modelo assim gerado pode crescer demasiadamente em alguns casos, levando a problemas de falta de memória ou consumindo tempo de computação excessivo. Por isso, implementamos uma variação do algoritmo que, a cada iteração, dada a solução atual, verifica as folgas das restrições (3) já adicionadas ao modelo. Se a folga for maior que uma certa constante, removemos a restrição do modelo.

Implementamos também versões similares desses dois algoritmos com a formulação $mdf-MST_{SEC}$, dado que as restrições de eliminação de ciclo também podem ser separadas em tempo polinomial (Magnanti; Wolsey, 1994). Resolvemos ainda a formulação $mdf-MST_{MTZ}$ no CPLEX. Nesse caso, como o número de variáveis e restrições é polinomial, a formulação é diretamente submetida ao *solver*.

Os resultados computacionais com estas duas últimas formulações foram bem inferiores àqueles obtidos com $mdf-MST_{cutsetD}$, motivo pelo qual detalhamos somente estes na próxima seção.

7 Experimentos Computacionais

No trabalho de Martins e Souza (2009), foram considerados três grupos de instâncias de testes, denominadas *CRD*, *SYM* e *ALM*, definidas por grafos completos e com custos positivos nas arestas. Todas são definidas no plano Euclidiano. Realizamos nossas experiências computacionais com todas as instâncias, mas, por limitação de espaço, apresentamos aqui apenas resultados com o grupo *ALM*, que compreende os maiores grafos, com 100, 200, 300, 400, 500 vértices. Fizemos os testes com $d = 3$ e $d = 5$. Como sugere a Proposição 17, o problema torna-se fácil para valores grandes de d . Consideramos a quantidade de centrais conforme a Tabela 1, de modo a respeitar a Proposição 1 e garantir a viabilidade.

# Vertices	$d = 3$	$d = 5$
	# Centrais	# Centrais
100	20, 30, 40 e 49	20 e 24
200	40, 50, 60, 70, 80, 90 e 99	20, 30, 40 e 49
300	100, 110, 120, 130 e 140	40, 50, 60, 70 e 74
400	100, 110, 120, 130, 140, 150, 160, 170, 180 e 190	40, 50, 60, 70, 80, 90 e 99
500	100, 120, 140, 160, 180, 200, 220 e 240	100, 110, 120 e 124

Tabela 1: Instâncias de Teste ALM

Utilizamos essas instâncias para avaliar o desempenho das formulações e algoritmos. Decidimos usar a versão 11.1 do CPLEX, em vez da versão 12.2, porque aquela apresentou, em todas as instâncias testadas, menores tempos computacionais. Os testes foram feitos na linguagem C, no sistema operacional Linux com um computador Intel Core i5, com 2.53 GHz e 4 GBytes de memória RAM.

Nas tabelas abaixo mostramos os resultados para $d = 3$. Na coluna RL, temos o valor da relaxação linear quando sua solução não é inteira. As colunas T1 e T2 apresentam os tempos do algoritmo quando não removemos e removemos as restrições folgadas do tipo (3), respectivamente.

Observamos que, das 102 instâncias testadas para $d = 3$, apenas em 39 delas a solução da relaxação linear não foi inteira, o que mostra a robustez da formulação. Dessas 39 instâncias, 29 delas foram depois resolvidas à otimalidade com o branch-and-bound do CPLEX em sua parametrização padrão. As demais, todas com 500 vértices, não puderam ser resolvidas no tempo limite de 2 horas ou devido à falta de memória. Para $d = 5$, todas as 66 instâncias testadas foram resolvidas à otimalidade, sendo 57 delas já pela relaxação linear.

A remoção das restrições não ativas do modelo quase sempre levou a uma redução (significativa) do tempo computacional, à exceção de pouquíssimas instâncias, quase todas de pequeno porte. Além disso, possibilitou resolver várias instâncias para as quais anteriormente ocorria falta de memória ou extrapolação do tempo limite.

Problema	# Centrais	RL	Solução Ótima	T1(s)	T2(s)
Tb8ch100.0	20		6641	0,03	0.03
	30		5895	0,09	0.09
	40		5795	0,18	0.15
	49	5756	5777	0,52	0.49
Tb8ch100.1	20		6740	0,04	0.03
	30		5591	0,07	0.06
	40	5617	5639	0,55	0.53
	49	5673	5676	0,81	0.65
Tb8ch100.2	20		7553	0,02	0.02
	30		6283	0,12	0.13
	40		5801	0,49	0.47
	49		5721	0,26	0.29

Tabela 2: Resultados do $\text{mdf-MST}_{\text{cutset}D}$ para $n=100$ e $d=3$

Problema	# Centrais	RL	Solução Ótima	T1(s)	T2(s)
Tb8ch200_0	40		9205	0,76	0.95
	50		8299	2,53	2.58
	60		7951	6,46	6.37
	70		7586	15,53	11.73
	80		7136	32,65	19.95
	90	7054,5	7055	76,47	41.86
	99		7980	4,91	4.41
Tb8ch200_1	40		9978	0,82	0.93
	50		8944	1,04	1.02
	60		8226	2,86	2.67
	70		7929	4,74	4.19
	80	7842,5	7843	37,22	22.43
	90	7897	7897	49,39	33.40
	99	8328	8331	39,36	33.25
Tb8ch200_2	40		9971	1,67	1.27
	50		8740	3,42	3.95
	60		8115	8,15	8.03
	70		7701	15,82	11.97
	80	7630	7630	23,73	18.61
	90	7480,5	7482	35,53	24.85
	99	7698	7723	80,61	59.07

Tabela 3: Resultados do $\text{mdf-MST}_{\text{cutset}D}$ para $n=200$ e $d=3$

Problema	# Centrais	RL	Solução Ótima	T1(s)	T2(s)
Tb8ch300_0	100	9795,5	9796	149	143.42
	110		9272	408	272.21
	120		9229	472	460.99
	130	9002,5	9006	1241	1145
	140	8979	8986	1031	818.48
	Tb8ch300_1	100		8985	422
110			8693	860	494.90
120			8398	989	569.69
130		8492	8492	685	619.01
140		8419	8420	1708	956.21
Tb8ch300_2	100		9319	131	86.31
	110	9069,5	9070	271	174.32
	120	9097,5	9098	631	464
	130	9049,5	9051	684	364.59
	140		9194	617	392.86

Tabela 4: Resultados do $\text{mdf-MST}_{\text{cutset}D}$ para $n=300$ e $d=3$

Problema	# Centrais	RL	Solução Ótima	T1(s)	T2(s)
Tb8ch400_0	100		12101	264	291.84
	110		11544	380	294.61
	120		11307	772	499.52
	130		10995	2067	972.74
	140		10711	1962	1096.16
	150		10433	3610	1829.04
	160	10316	10322	3125	1305.52
	170		10092	> 2 horas	3619
	180	10070,25	10075	sem memoria	3699
	190	10455,66667	10459	sem memoria	3462
Tb8ch400_1	100	12739	12740	174	166.24
	110	12361	12362	618	286
	120	11837	11838	1659	931.42
	130	11391	11393	sem memoria	2837

	140	11000	11002	sem memoria	3379.61
	150	10856	10858	sem memoria	3828
	160	10713	10717	> 2 horas	4546
	170	10634	10637	> 2 horas	3953
	180	10676	10680	> 2 horas	6429
	190	10918.50	10923	> 2 horas	5421
Tb8ch400_2	100		13043	124	74.94
	110		12360	131	91.20
	120		11933	486	247.12
	130		11603	462	236.20
	140	11484,5	11486	708	339.25
	150	10984	10985	883	413.99
	160	10725	10726	1754	771.76
	170	10646	10646	1831	666.04
	180	10511	10511	sem memoria	1341
	190	10742	10746	1330	600.80

Tabela 5: Resultados do mdf-MST_{cutsetD} para n=400 e d=3

Problema	# Centrais	RL	Solução Ótima	T1(s)	T2(s)
Tb8ch500_0	100		14735	212	152.42
	120		13463	777	388.88
	140		12457	4352	2371
	160		11969	> 2 horas	5150
	180		11682	> 2 horas	8251
	200			> 2 horas	> 2 horas
	220			sem memoria	> 2 horas
	240			sem memoria	sem memoria
Tb8ch500_1	100		15912	194	111.92
	120		14198	1146	732.93
	140		13298	2632	2180.55
	160		12693	6402	3330
	180		12049	> 2 horas	6317
	200			sem memoria	> 2 horas
	220			sem memoria	sem memoria
	240			sem memoria	> 2 horas
Tb8ch500_2	100		14046	147	156.33
	120	12940	12940	773	681
	140	12390	12390	3245	2278
	160			> 2 horas	> 2 horas
	180		11647	> 2 horas	5813
	200			sem memoria	> 2 horas
	220			sem memoria	> 2 horas
	240			sem memoria	> 2 horas

Tabela 6: Resultados do mdf-MST_{cutsetD} para n=500 e d=3

8 Conclusão e Perspectivas

Apresentamos uma variação do problema de Árvore Geradora Mínima com Restrição de Grau Mínimo onde os vértices centrais e terminais são definidos a priori. Mostramos que o problema é NP-Difícil para qualquer valor do parâmetro $d \geq 2$ ou mesmo que cada terminal se ligue a todos os centrais. Apontamos que diferentes formulações de programação inteira podem ser obtidas tomando como base descrições de árvores geradoras. Implementamos três formulações para o problema, sendo que em duas delas empregamos algoritmos clássicos de separação de restrições para resolvê-las. A formulação direcionada de eliminação de ciclos mostrou-se ser largamente melhor que as outras duas, demandando tempo computacional bem menor. Das 168 instâncias testadas, em 120 a solução da relaxação linear já é inteira e, em apenas 10 delas, todas com 500 vértices, não encontramos a solução ótima, seja devido

à limitação de memória ou tempo. Esse resultado mostra que essa formulação é bastante forte.

Para melhorar o desempenho com as instâncias maiores, estamos desenvolvendo várias heurísticas para incorporar ao branch-and-bound, entre elas heurísticas gulosas, procedimentos VNS/VND e heurísticas adaptadas do md-MST. Paralelamente, estamos desenvolvendo um branch-and-bound específico, explorando outras regras de ramificação e limites inferiores de boa qualidade e baixo custo computacional.

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, 10:51-71, 1994.
- [2] **Almeida, A. M.; Martins, M.; Souza, M.C.** *Min-Degree Constrained Minimum Spanning Tree Problem: Complexity, properties and formulations.* Technical Report 6/2006, Centro de Investigação Operacional, Universidade de Lisboa, 2006.
- [3] **Dantzig, G. B.; Fulkerson, D. R. e Johnson, S. M.** *Solution of a large scale traveling salesman problem.* Operations Research, 2:393-410, 1954.
- [4] **Esau, L.R.; Williams, K.C.** *On teleprocessing network design: Part II. A method for approximating the optimal network.* IBM Systems Journal 5 (3): 142-147, 1966.
- [5] **Fürer, M.; Raghavachari, B.** *Approximating the minimum-degree Steiner tree to within one of optimal,* Journal of Algorithms 17 (3): 409-423, 1994.
- [6] **Garey, M.R.; Johnson, D.S.** *Computers and Intractability: A Guide to the Theory of NP-Completeness,* A2.1: ND1, p. 206, 1979.
- [7] **Gouveia, L.** *Multicommodity Flow Models for Spanning Trees with Hop Constraints.* European Journal of Operational Research, 95:178-190, 1996.
- [8] **Martins, P.; Souza, M. C.** *VNS and second order heuristics for the min-degree constrained minimum spanning tree problem.* Computers and Operations Research, 36:2669-2982, 2009.
- [9] **Martinez, L.C.; Cunha, A.** *The Min-Degree Constrained Minimum Spanning Tree Problem: Formulations and Branch-and-cut algorithm.* Discrete Applied Mathematics, 2013. doi: 10.1016/j.dam.2011.08.008. In Press.
- [10] **Martinez, L.C. ; Cunha, A.** *A Parallel Lagrangian Relaxation Algorithm for the Min-Degree Minimum Spanning Tree Problem.* Lecture Notes in Computer Science - Combinatorial Optimization. v. 7422, p. 237-248, 2012.
- [11] **Miller, C; Tucker, A; Zemlin, R.** *Integer programming formulation of traveling salesman problems.* Journal of ACM, 7:326-9, 1960.
- [12] **Myung, Y., Lee, C., Tcha, D.** *On the generalized minimum spanning tree problem.* Networks, 26(4), 231-241, 1995.
- [13] **Narula, S. C.; Ho, C. A.;** *Degree-constrained minimum spanning tree.* Computers and Operations Research, 7:239-249, 1980.
- [14] **Salama, H.F., Reeves, D.S., Viniotis, Y.** *The Delay-Constrained Minimum Spanning Tree Problem.* In Proceedings of the 2nd IEEE Symposium on Computers and Communications (ISCC '97). IEEE Computer Society, Washington, DC, USA, 1997.
- [15] **Magnanti, T.L. and Wolsey, L.A.** *Optimal Trees* OR 290-94 April 1994.