LIII Simpósio Brasileiro de Pesquisa Operacional
João Pessoa - PB, 3 a 5 de novembro de 2021

Métodos Exatos para o Problema da Árvore de Cobertura com Representação Mínima

Victor Koehler[2], David Galvão[2], Thiago Gouveia[1], Gilberto Farias de Sousa Filho[2]
[1]Instituto Federal da Paraíba, João Pessoa — PB
[2]Universidade Federal da Paraíba, João Pessoa — PB
João Pessoa — PB
Autor para correspondência: koehler.victor@gmail.com

RESUMO

Neste trabalho abordamos o Problema da Árvore de Cobertura com Representação Mínima (PACRM), um problema relativamente recente na literatura acadêmica. Neste, dado um grafo com arestas rotuladas $G = (V, E, L)$, sendo V o conjunto de vértices, E o conjunto de arestas, L o conjunto de rótulos, e cada aresta $e \in E$ possui um rótulo $L(e)$ associado, o objetivo é encontrar uma árvore geradora $T = (V, E', L')$, tal que $E' \subseteq E, L' \subseteq L$, e a soma dos rótulos representados em cada vértice seja minimizada. Propomos dois métodos exatos para o problema, um algoritmo de refinamento do grafo de entrada, algoritmos de *branch-and-cut* e novas desigualdades válidas. Os experimentos computacionais realizados demonstraram que os novos modelos matemáticos encontraram mais soluções ótimas e em menor tempo comparadas ao estado da arte.

PALAVRAS CHAVE. Grafos com Arestas Rotuladas, Árvores, Branch-and-Cut.

Área Principal: PM - Programação Matemática

ABSTRACT

In this work we address the Minimum Representation Spanning Tree Problem (MRSTP), a relatively recent problem in the scientific literature whose goal is: given an edge-labeled graph $G = (V, E, L)$, where V is the set of vertices of G, E is the set of edges, L is the set of labels (colors), and each edge $e \in E$ has a label $L(e)$ associated, find a spanning tree $T = (V, E', L')$, such that $E' \subseteq E$, $L' \subseteq L$ that minimizes the sum of the labels represented in each vertex. We propose two exact methods for the problem, a refinement algorithm for the input graph, branch-and-cut algorithms, and new valid inequalities. The computational experiments have shown that the new mathematical models were able to find more optimal solutions in less time compared to the state-of-the-art.

KEYWORDS. Edge-Labeled Graphs, Trees, Branch-and-Cut.

Main Area: MP - Mathematical Programming

1. Introdução

Um grafo é uma estrutura de dados abstrata composta por um conjunto de vértices e por um conjunto de arestas, cada uma representando uma ligação entre dois vértices. Por sua vez, uma árvore é um tipo especial de grafo no qual, tomando-se quaisquer par de vértices distintos, há exatamente um caminho entre estes. Uma característica importante das árvores é que estas representam o menor grafo conexo possível em relação ao número de arestas, uma que vez que remover qualquer aresta desconectará o grafo, enquanto que adicionar uma aresta entre a e b criará um segundo caminho entre estes dois vértices. Por este motivo, é muito comum utilizar o conceito de árvore para modelar problemas de conectividade, como no Problema da Árvore de Steiner (Chen e Hsieh, 2020), no Problema da Árvore-Estrela (Gurguri e Andrade, 2017) e no Problema da Árvore Geradora Mínima sob Restrições de Conflitos (Barbosa e Delbem, 2017).

Por outro lado, um grafo com arestas rotuladas (GAR) é um grafo no qual cada aresta possui um rótulo associado, em vez de um peso. Nesta classe de grafos, é comum utilizar cores para melhorar a visualização dos rótulos, e, por este motivo, os termos rótulos e cores são utilizados de forma intercambiável neste trabalho. Nos últimos anos, diversos trabalhos foram realizados sobre problemas relacionados a GARs. Destes, podemos ainda relacionar aqueles que buscam por árvores, como o Problema da Árvore Geradora com Rotulação Mínima (Silva et al., 2018), o Problema de Steiner de Rotulação Mínima (Cerulli et al., 2006) e a versão Fortemente Generalizada do Problema da Árvore de Cobertura de Rotulação Mínima (Cerrone et al., 2019).

Este trabalho aborda o Problema da Árvore de Cobertura com Representação Mínima (PACRM, (Silva, 2018; Linares et al., 2018)). Neste, é dado um grafo com arestas rotuladas como entrada, e objetiva-se encontrar uma árvore de cobertura que minimize a soma das cores representadas em cada vértice. O PACRM possui aplicações diretas no projeto de redes de computadores, no qual é desejável que cada nó possua a menor complexidade possível, o que torna o projeto mais barato, além de facilitar a sua manutenção. Por exemplo, considere o diagrama lógico de um projeto de rede local apresentado na Figura 1. Neste, é possível conectar os *switches* usando cabos Ethernet, em azul, ou de fibra óptica, em laranja. *Switches* apenas com capacidade Ethernet são mais baratos, assim como ativos apenas ópticos. Combinar as duas tecnologias torna o equipamento mais caro, pois requer a transformação do sinal. No diagrama da Figura 1, é possível conectar todos os nós usando dois *switches* apenas ópticos ($S5$ e $S9$), oito *switches* apenas Ethernet ($S2$, $S3$, $S4$, $S6$, $S7$, $S8$, $S10$ e $S11$), e apenas um *switch* com ambas capacidades ($S1$).

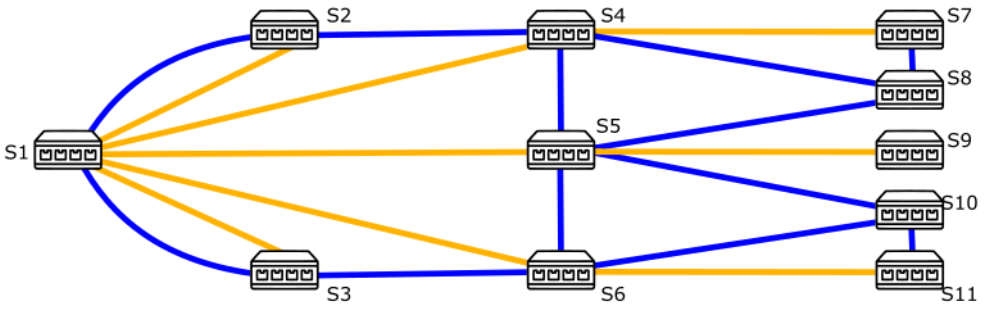


Figura 1: Exemplo de diagrama lógico para projeto de rede local

Formalmente, considere $G = (V, E, L)$ um GAR simples e não orientado, sendo V o conjunto de vértices, E o conjunto de arestas, L o conjunto de rótulos (ou cores), e $L(e)$ uma função que retorna o rótulo associado à aresta $e \in E$. Seja $E(v)$ o conjunto de arestas que incidem

sobre o vértice v, $c_T(v) = \bigcup_{e \in E(v)} L(e)$ o conjunto de rótulos que incidem sobre o vértice v, e $Rt(G) = \sum_{v \in V}(|c_T(v)| - 1)$ a função de *representação*. O objetivo do PACRM é encontrar uma árvore geradora $T = (V, E', L')$, tal que $E' \subseteq E$, $L' \subseteq L$ e $Rt(T)$ é minimizado. A Figura 2 detalha os conceitos de GAR e de representação, assim como exemplifica o Problema da Árvore de Cobertura com Representação Mínima. A Figura 2(a) ilustra um GAR $G = (V, E, L)$ com 6 vértices, 10 arestas e 6 rótulos. A Figura 2(b) exibe uma árvore de cobertura T com $Rt(T) = 2$, menor valor possível, e 3 cores. Em detalhes, os vértices 1, 3, 4 e 6 possuem $|c_T(v)| = 1$, enquanto os vértices 2 e 5, marcados da figura, possuem $|c_T(v)| = 2$. Logo, $Rt(G) = \sum_{v \in V}(|c_T(v)| - 1) = 2$. Adicionalmente, observe que a árvore de cobertura de G com o menor número de cores utiliza apenas o conjunto de rótulos $\{A, B\}$.

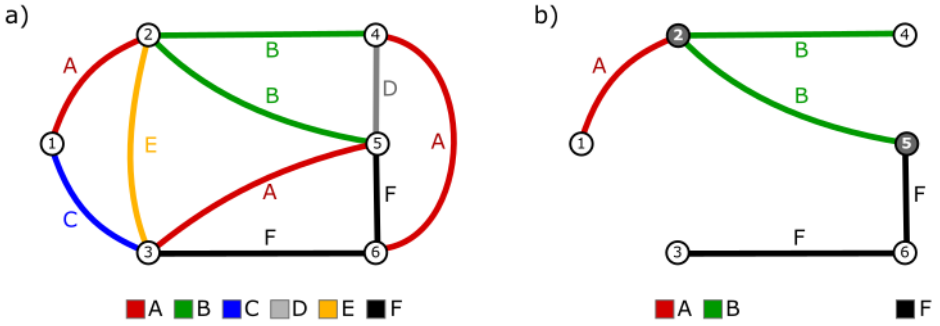


Figura 2: (a) Um pequeno GAR G como Instância-exemplo para o PACRM, (b) Solução ótima para o PACRM em G com valor $Rt(T) = 2$

Neste trabalho, apresentamos um refinamento da transformação do PACRM para o Problema de Steiner com Rotulação Mínima proposta por Linares et al. (2018), dois métodos exatos para a resolução do PACRM, seus respectivos algoritmos de *branch-and-cut* e novas desigualdades válidas. As demais seções deste trabalho estão organizadas como segue: a Seção 2 apresenta trabalhos relacionados ao tema; a Seção 3 detalha as transformações polinomiais do PACRM; a Seção 4 apresenta os métodos exatos propostos; os experimentos computacionais são discutidos na Seção 5; enquanto que as considerações finais e propostas de trabalhos futuros são trazidas na Seção 6.

2. Trabalhos Relacionados

Um problema amplamente estudado na literatura e fortemente relacionado ao PACRM é o Problema da Árvore de Cobertura com Rotulação Mínima (MLST, (Broersma e Li, 1997; Silva, 2018)) que consiste em: dado um GAR $G = (V, E, L)$, encontrar uma árvore de cobertura rotulada $T = (V, E', L')$ que minimize a quantidade de rótulos $|L'|$ utilizados. Chen et al. (2008) propõem uma extensão a esse problema denominado Problema da Árvore de Cobertura com Rotulação Mínima Generalizada (GMLST), no qual cada aresta possui um ou mais rótulos associados, e a utilização dessas arestas na solução implica na utilização de um subconjunto não vazio desses mesmos rótulos. Por sua vez, Cerrone et al. (2019) apresentam ainda outra extensão sobre o GMLST, o Problema da Árvore de Cobertura com Rotulação Mínima Fortemente Generalizada (SGMLST), no qual o uso de uma aresta na solução resulta na utilização de todos os seus rótulos.

Proposto por Cerulli et al. (2006), o Problema de Steiner de Rotulação Mínima (MLSteiner), consiste em: dado um GAR $G = (V, Q, E, L)$ onde $Q \subseteq V$ é o subconjunto de vértices *terminais*, minimizar um subconjunto de rótulos $L' \subseteq L$ de modo a encontrar um subgrafo conexo $G' = (V', E', L')$ tal que Q esteja inteiramente contido em V'. Ademais, definimos como *vértice*

de Steiner todo $s \in S,\ S = V \setminus Q$. Informalmente, G' é um subgrafo que possui todos os vértices de Q conectados, utilizando opcionalmente alguns vértices de Steiner no processo, objetivando minimizar o conjunto de cores necessárias para tal fim. Os autores propuseram ainda uma heurística gulosa, meta-heurísticas baseadas em Busca Tabu, *Simulated Annealing*, *Variable Neighborhood Search* e, por fim, uma meta-heurística baseada no *Pilot Method*, que obteve os melhores resultados nos experimentos realizados. Por fim, Consoli et al. (2009) propõem um método exato e uma heurística baseada na técnica de Enxame de Partículas denominada *JPSO*, o qual representa o estado da arte para o MLSteiner.

Silva (2018) propõe o PACRM e demonstra que o problema é NP-Completo tanto para multigrafos quanto para grafos simples. Além disso, apresenta uma formulação matemática baseada em cortes de arestas para o problema. Este modelo, denominado EDGE, é apresentado nas expressões (1 - 5).

$$\text{Minimize}\Big(\sum_{v \in V}\sum_{l \in L} z_l^v\Big) - |V| \tag{1}$$

$$\text{s.t.} \sum_{x_e \in \delta(S)} x_e \geq 1, \qquad \forall S \subset V, S \neq \emptyset, \tag{2}$$

$$z^v_{L(e)} \geq z^w_{L(e)} \geq x_e, \qquad \forall e = (v,w) \in E, \tag{3}$$

$$\sum_{e \in E} x_e = |V| - 1, \tag{4}$$

$$z_l^v, x_e \in \{0,1\}, \qquad \forall l \in L, \forall v \in V, \forall e \in E. \tag{5}$$

Cada variável binária $z_l^v, \forall v \in V, \forall l \in L$, assume o valor 1 se e somente se a cor l incide sobre o vértice v. Por sua vez, cada variável binária x_e, $\forall e \in E$, assume o valor 1 se e somente se a aresta e faz parte da solução. A função objetivo (1) minimiza a representação da solução; o conjunto de desigualdades (2) garante que o grafo solução é conexo, onde $\delta(S) = \{e = (v,w) \in E : |\{v,w\} \cap S| = 1\}$; o conjunto de restrições (3) ativa os rótulos da aresta $e = (v,w)$ caso $x_e = 1$; enquanto que a equação (4) garante que a solução será uma árvore. Por fim, o domínio das variáveis é definido em (5).

Além disso, Silva (2018) propôs duas heurísticas construtivas gulosas para o problema, denominadas *KBased* e *PBased*, inspiradas nos algoritmos de Prim e Kruskal (Cormen et al., 2009), respectivamente. Considere $G = (V, E, L)$ o GAR de entrada, e $\mathcal{K}$ o conjunto de componentes monocromáticas maximais conexas (CMMC) de G, construído com o auxílio do algoritmo de busca em profundidade aplicado sobre cada subgrafo induzido $G = (V, E(\{l\}), \{l\}), \forall l \in L$, onde $E(\{l\}) = \{e, \forall e \in E \mid L(e) = l\}$.

Heurística *KBased*: Inicia com o conjunto de arestas solução $E' = \emptyset$ e, enquanto $G(V, E', L)$ não for conexo, encontre o CMMC $K \in \mathcal{K}$ que minimize o número de componentes conexas de $G(V, E', L)$ e adicione em E' as arestas de K que não formem ciclo.

Heurística *PBased*: Inicia o conjunto de arestas solução $E' = \emptyset$ e $S = \{r\}$, onde $r \in V$ é um nó raiz escolhido arbitrariamente. Enquanto $S \neq V$, encontre o CMMC $K \in \mathcal{K}$ que minimize o incremento da representação mínima dos vértices de S, adicione os vértices de K em S sem repetição e adicione em E' as arestas de K que não formem ciclo.

Adicionalmente, Linares (2018) demonstra que o PACRM permanece NP-Completo mesmo no caso de $|L| = 2$, isto é, mesmo se o grafo de entrada possuir apenas duas cores. E propõe um método baseado na meta-heurística GRASP que utilizou um mapeamento (detalhado na Seção 3) do PACRM para o Problema de Steiner de Rotulação Mínima e uma adaptação da heurística MVCA, sugerindo o uso dessas heurísticas para instâncias de média e alta densidades e de métodos exatos para instâncias de baixa densidade e poucos vértices.

3. Transformações Polinomiais

Nesta seção, introduzimos novos conceitos utilizados no decorrer do trabalho e descrevemos uma transformação polinomial sobre a instância de entrada do PACRM com o objetivo de resolvê-lo através de sua redução ao Problema de Steiner de Rotulação Mínima. Em seguida, apresentamos um segundo algoritmo para transformar o grafo de entrada, desta vez em uma nova variante do Problema de Steiner tomando como base as características do Problema da Árvore de Cobertura de Rotulação Mínima Fortemente Generalizada.

3.1. MLSteiner

Linares et al. (2018) definiram uma transformação aplicável a toda instância do PACRM para o MLSteiner, a qual iremos reintroduzir com a notação deste trabalho a fim de simplificar as formulações propostas nas próximas seções. Seja $G = (V, E, L)$ o GAR de entrada para o PACRM, define-se $G_{Steiner} = (V', V, E', L')$ como um grafo-problema para o MLSteiner, onde $V' = V \cup \{ s_e, \forall e \in E \}$ é a união dos vértices de V com um novo *vértice de Steiner* para cada aresta de E; $E' = \{ (v, s_e), (w, s_e) \mid \forall e = (v, w) \in E \}$ é constituído pela transformação de cada aresta de E para outras duas que conectam o *vértice de Steiner* gerado em V' aos vértices originais v e w; por fim, seja $K(e') = (v, L(e)),\ e' = (v, s_e) \in E', e \in E$, então $L' = \{ K(e') \mid e' \in E' \}$ é o conjunto de rótulos que conjuga o par vértice-cor unicamente, de forma que uma mesma cor pode estar presente várias vezes no conjunto L', desde que associada a diferentes vértices, ao passo que $K(e')$ é uma função que mapeia arestas de E' para rótulos de L'.

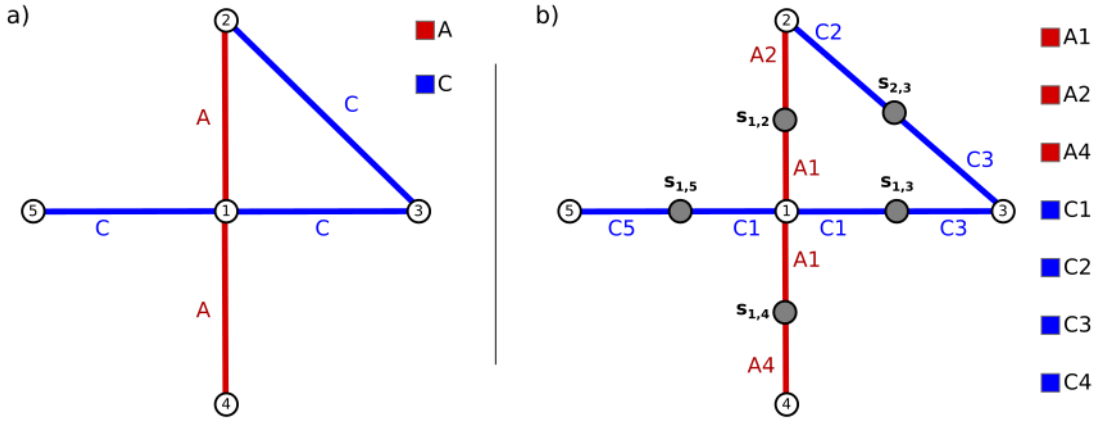


Figura 3: (a) Um pequeno GAR G como Instância-exemplo para o PACRM, (b) Transformação de G em uma instância do MLSteiner.

A Figura 3 ilustra a transformação do grafo (a) para uma instância do MLSteiner em (b). Observa-se que todas as arestas são transformadas em duas arestas de mesma cor, conectando os vértices originais por meio de um novo vértice de Steiner, representado por circunferências de cor cinza. Além disto, é possível notar que, embora as cores das arestas permaneçam as mesmas, elas

estão rotuladas junto com o vértice terminal (não-Steiner) que as deu origem. Consequentemente, as arestas $(3, s_{1,3})$ e $(3, s_{2,3})$ possuem estritamente o mesmo rótulo, diferentemente de $(3, s_{1,3})$ e $(1, s_{1,3})$ que, embora possuam a mesma cor na figura, possuem estritamente rótulos diferentes, isto é, $K((3, s_{1,3})) = (3, C) \neq K((1, s_{1,3})) = (1, C)$.

3.2. SGMLSteiner

Adicionalmente, propomos um algoritmo que visa refinar o grafo de Steiner de entrada e evidenciar propriedades úteis para o modelo. Para tanto, usamos o Problema da Árvore de Cobertura de Rotulação Mínima Fortemente Generalizada (SGMLST, do inglês *Strong Generalized Minimum Label Spanning Tree Problem*), uma variante do MLST proposta por Cerrone et al. (2019) na qual cada aresta pode possuir um ou mais rótulos tal que, se usada na solução, todos os seus rótulos são contabilizados. Formalmente, seja um GAR $G = (V, E, L)$, V o conjunto de vértices, E o conjunto de arestas, L o conjunto de rótulos e $\mathcal{L}(e), e \in E$, uma função que mapeia cada aresta para um subconjunto de cores, o SGMLST objetiva encontrar uma árvore $T = (V, E', L')$, $E' \subseteq E$, $L' \subseteq L$, tal que $l \in L', \forall l \in \mathcal{L}(e), e \in E'$ e $|L'|$ seja minimizado.

Em seguida, definimos o Problema de Steiner de Rotulação Mínima Fortemente Generalizado (SGMLSteiner) como uma combinação das características do MLSteiner e do SGMLST. Formalmente, seja um GAR $G = (V, Q, E, L)$, onde $Q \subseteq V$, e $\mathcal{L}(e), \forall e \in E$, é uma função que mapeia cada aresta para um subconjunto de cores, o objetivo do SGMLSteiner é encontrar um subgrafo conexo $G' = (V', E', L')$ tal que $|L'|$ seja minimizado, Q esteja inteiramente contido em V' e todo rótulo l de cada aresta da solução esteja em L'. Finalmente, utilizaremos o SGMLSteiner através do Algoritmo 1, que recebe um GAR e retorna a sua transformação $G' = (V', V, E', L')$, que admite um ou mais rótulos para uma mesma aresta.

A transformação proposta pelo Algoritmo pode ser descrita em três casos gerais de tratamento: (i) se um vértice v possui duas ou mais arestas de cor l incidentes, então um vértice de Steiner é inserido no grafo para agrupá-las; (ii) se houver uma única aresta e o vértice w oposto a ela também pertencer a este caso, a aresta é mantida com apenas uma cor; do contrário, (iii) a aresta é atualizada de modo a conter os rótulos conjugados (v, l) e (w, l). Na Figura 4, ambos os vértices 2 e 3 correspondem ao segundo caso em relação ao rótulo A. Em relação ao rótulo C, os vértices 1, 4 e 5 referem-se ao primeiro caso, enquanto 2 e 3 ao terceiro caso. Observa-se, porém, que todo rótulo do caso (ii) passa a possuir peso 2 na função representação.

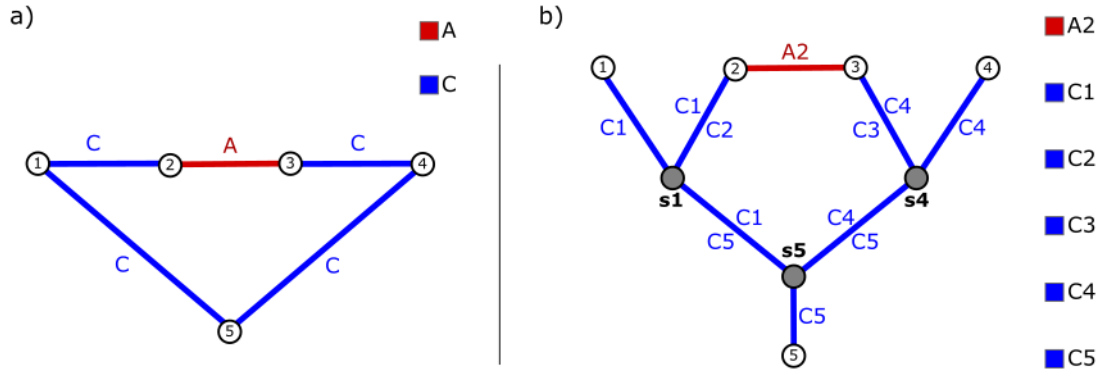


Figura 4: (a) Um pequeno GAR como Instância-exemplo para o PACRM, (b) O mesmo grafo representado pela transformação ao SGMLSteiner proposto.

A Tabela 1 relaciona as duas transformações apresentadas em comparação aos grafos de

Algoritmo 1: refinarGrafo($G = (V, E, L)$)

$V' \leftarrow V,\ E' \leftarrow E,\ L' \leftarrow \{\}$
Seja $K \leftarrow \{\}$ um mapeamento de E' para L'
Seja $V_m \leftarrow \{\}$ um mapeamento de V' para V
Para cada $v \in V$ **faça**
    $V_m(v) \leftarrow v \mid \forall v \in V$
    Para cada $e_{v,w} \in E' \mid \forall w \in V', \exists e_{v,w} \in E'$ **faça**
        $K(e_{v,w}) \leftarrow \{ (v, L(e_{v,w})), (w, L(e_{v,w})) \}$
Para cada $v \in V$ **faça**
    Para cada $k \in L$ **faça**
        $Arestas_v^k \leftarrow \{ e_{v,w} \mid \forall w \in V', \exists e_{v,w} \in E',\ (v,k) \in K(e_{v,w})\}$
        Se $|Arestas_v^k| \geq 2$ **então**
            Seja v_{st} um novo vértice
            $V' \leftarrow V' \cup \{v_{st}\}$
            $E' \leftarrow E' \cup \{ e_{v,v_{st}} \}$
            $K(e_{v,v_{st}}) \leftarrow \{ (v,k) \}$
            $V_m(v_{st}) \leftarrow v$
            Para cada $e_{v,w} \in Arestas_v^k$ **faça**
                $E' \leftarrow E' \cup \{ e_{v_{st},w} \} \backslash \{ e_{v,w} \}$
                $K(e_{v_{st},w}) \leftarrow \{ (v,k), (V_m(w),k) \}$
Para cada $v, w \in V \mid \exists e_{v,w} \in E',\ (v,k) \in K(e_{v,w})$ **faça**
    $K(e_{v,w}) \leftarrow \{ (v,k) \}$
$L' = \{ K(e') \mid e' \in E' \}$
retorne $G_{SGMLSteiner} = (V', V, E', L')$

entrada. Para isso, GARs gerados de forma aleatória por Cerulli et al. (2005) foram utilizados. Cada linha representa a média de um grupo de 30 instâncias para cada $|L| = \{20, 30, 40, 50\}$. As colunas estão divididas em $Entrada$, $MLSteiner$ e $SGMLSteiner$, que representam, respectivamente, o grafo original de entrada adaptado ao PACRM e as transformações propostas. Essas se dividem em $|V'|$, $|E'|$ e $|L'|$, sendo, respectivamente, o número de vértices (Terminais e não Terminais), arestas e, por fim, o número absoluto dos pares vértices-rótulos conjugados presentes nas arestas. Ademais, convém ressaltar que as colunas de $Entrada$ possuem correspondência direta ao tamanho das variáveis do modelo EDGE, apresentado na Seção 2.

Tabela 1: Comparação entre as representações de grafos propostas para $|L| \in \{20, 30, 40, 50\}$

Inst.	**Entrada**			**MLSteiner**			**SGMLSteiner**		
$\|L\|$	$\|V'\|$	$\|E'\|$	$\|L'\|$	$\|V'\|$	$\|E'\|$	$\|L'\|$	$\|V'\|$	$\|E'\|$	$\|L'\|$
20	20.0	95.00	145.87	115.00	190.00	145.87	56.70	131.70	129.72
30	30.0	217.33	328.70	247.33	434.67	328.70	114.93	302.27	294.15
40	40.0	390.00	586.10	430.00	780.00	586.10	196.20	546.20	524.63
50	50.0	612.33	917.10	662.33	1224.67	917.10	293.40	855.73	823.13

Devido a transformação de cada aresta em outras duas, pode-se observar no $MLSteiner$ um aumento de 2 vezes no número de arestas e de 5 a 13 vezes no número de vértices em comparação com o grafo de $Entrada$, permanecendo igual apenas o conjunto de rótulos. Por sua vez, o $SGMLSteiner$ apresentou um aumento entre 3 a 6 vezes no número de vértices e, em média, 39% no número de arestas em comparação com $Entrada$, porém com uma redução de 11% em média no número de rótulos devido ao tratamento do segundo caso do Algoritmo 1.

4. Formulações Matemáticas

Nesta seção, são apresentados os modelos matemáticos propostos neste trabalho, assim como novas desigualdades válidas e algoritmos de *branch-and-cut* para resolução dos modelos.

4.1. Modelo PACRM-MLSteiner

O modelo de programação linear inteira apresentado a seguir constitui-se de um aprimoramento do modelo proposto por Silva (2018) para o MLSteiner e usa um conjunto exponencial de cortes que garantem a conectividade da solução. Seja $G_{Steiner} = (V', V, E', L')$ o grafo de Steiner obtido pela transformação descrita na Seção 3, $z_l^v, \forall (v,l) \in L'$ uma variável que assume valor 1 se e somente se o rótulo (v,l) está presente na solução. O modelo PACRM-MLSteiner é descrito por:

$$\text{Minimize} - |V| + \sum_{(v,l) \in L'} z_l^v \tag{6}$$

$$\text{s.t.} \sum_{(v,l) \in K(T)} z_l^v \geq 1, \qquad \forall T \subset V', T \cap V \neq \emptyset, T \cap V \neq V, \tag{7}$$

$$z_l^v \in \{0,1\}, \qquad \forall (v,l) \in L'. \tag{8}$$

A função objetivo (6) é a função representação Rt, isto é, o custo dos rótulos na solução; o conjunto de restrições (7) assegura a conectividade, sendo $K(T)$ o conjunto de cores incidentes no corte $\delta(T)$, e (8) define o domínio das variáveis de decisão.

Ressalta-se ainda que uma solução para o modelo PACRM-MLSteiner pode ser uma árvore, ou mesmo um grafo com 1 ou mais ciclos, devido a ausência de restrições de proibição de ciclos. Ainda assim, essas constituem-se soluções ótimas válidas para o PACRM, pois basta que, para todo ciclo da solução, remova-se qualquer aresta deste até que se obtenha uma árvore de mesmo custo do grafo-solução.

4.2. Desigualdades Válidas

Esta seção descreve algumas desigualdades inerentes à natureza do PACRM, as quais, apesar da garantia de otimalidade do modelo PACRM-MLSteiner, apresentam considerável impacto na performance do método. Seja $C(v,l) = \{(w,l) \in L' \mid \forall w \in V', \exists e^l = (v,w) \in E\}$ uma função que mapeia cada rótulo (v,l) para seus rótulos vizinhos (w,l), a expressão (9) diz que se há um rótulo l de um vértice v presente na solução, então há pelo menos um outro rótulo l nos vértices adjacentes a v também presente na solução, tal que esses dois ou mais vértices estão conectados por arestas desta cor.

$$\sum_{(w,l) \in C(v,l)} z_l^w \geq z_l^v, \qquad \forall (v,l) \in L', \tag{9}$$

Em seguida, introduziremos no modelo um conjunto de restrições e variáveis correspondentes a arestas, cujo objetivo é acelerar a convergência do resolvedor para soluções que se assemelhem a uma árvore. Para simplificar a notação, note que as seguintes formulações utilizam o

conjunto E, estando, portanto, desassociadas a sua respectiva representação no grafo de Steiner obtido pelas transformações da Seção 3.

A expressão (10) diz que uma solução válida para o problema deve possuir exatamente $|V| - 1$ arestas; enquanto (11) associa cada aresta a seus respectivos rótulos no modelo, tal que se a aresta está presente no corte, os rótulos também devem estar; e (12) define o domínio das variáveis.

$$\sum_{e \in E} x_e = |V| - 1, \tag{10}$$

$$z^v_{L(e)} \geq x_e, \qquad \forall v \in W(e), e \in E, \tag{11}$$

$$x_e \in \{0, 1\}, \qquad \forall e \in E \tag{12}$$

Atenta-se, novamente, que o modelo não proíbe a ocorrência de ciclos, mesmo com a inclusão destas desigualdades, e que nesses casos, é possível que as variáveis x assumam, isoladamente, valores que não representem uma solução válida e ótima para o problema. Entretanto, ressalva-se que, mesmo diante destas ocorrências, os valores das variáveis z são suficientes para obter a representação de uma solução válida.

4.3. Algoritmo de branch-and-cut

Devido a quantidade exponencial de restrições da expressão (7), optou-se por inseri-las dinamicamente durante a resolução do modelo. Dessa forma, as restrições são inicialmente relaxadas e, para cada solução inteira encontrada durante o processo de *branch-and-bound*, verifica-se se estão violadas e insere-as da seguinte forma:

Seja $(\bar{z})$ uma solução inteira encontrada durante a otimização, definimos que uma aresta $e^l = (v, w) \in E'$ está presente na solução se e somente se $\bar{z}^v_l = \bar{z}^w_l = 1$. Dessa forma, aplica-se um algoritmo de busca em largura até que se obtenha uma floresta F que contenha todos os vértices de G. Em seguida, para cada árvore $T \in F$ que contenha pelo menos um e no máximo $|V| - 1$ vértices de V, obtemos e inserimos no corte as arestas que conectam T a outras árvores, isto é, $e^l = (v, w) \in E'$, tal que $\bar{z}^v_l + \bar{z}^w_l \leq 1$, para dois vértices v e w, de forma que apenas um deles esteja em T. Formalmente:

$$\sum_{(v,l) \in K(T)} z^v_l \geq 1, \qquad \forall T \in F, T \cap V \neq \emptyset, T \cap V \neq V, \tag{13}$$

e $K(T) = \{\ (v, l) \in L' \ \mid\ \forall v, w \in V',\ \exists e^l = (v, w) \in E,\ \bar{z}^v_l = 0,\ |\{v, w\} \cap T| = 1\ \}$.

5. Experimentos computacionais

Os métodos estudados neste trabalho foram desenvolvidos em linguagem C++, utilizando o compilador g++ versão 5.4.0 (opção de compilação -O2), com o auxílio do resolvedor matemático CPLEX versão 12.9 para os métodos exatos. Os experimentos computacionais foram executados em um computador com processador Intel Xeon E5-2650 v4 (64 bits) com 12 núcleos de 2,20 GHz, além de 128 GB de RAM e executando o sistema operacional Ubuntu Linux 16.04. Os experimentos foram realizados com 1 *thread* e tempo limite de uma hora.

Para os experimentos realizados, GARs gerados de forma aleatória por Cerulli et al. (2005) foram utilizados. Para cada $n = |V| = \{20, 30, 40, 50\}$, as instâncias possuem 30 grafos-problema, que são divididos em 3 grupos definidos pelos valores de $d = \{ld = 20\%; md = 50\%; hd = 80\%\}$, representando a densidade do grafo. Cada grupo possui 10 instâncias distintas de mesma dimensão, totalizando 120 grafos-problema utilizados nos experimentos computacionais.

Implementamos e comparamos 3 modelos matemáticos e suas estratégias de *branch-and-cut*. O primeiro modelo foi proposto por Silva (2018), aqui intitulado *EDGE*, os outros dois são modelos baseados em cortes coloridos para o problema MLSteiner, proposto na Seção 4, que recebem os grafos de entrada do PACRM reduzidos para o problema MLSteiner. Estas reduções estão descritas na Seção 3 e dão nome as estratégias analisadas *MLSteiner* e *SGMLSteiner*.

A Tabela 2 apresenta a comparação dos modelos para os conjuntos de vértices de tamanho $|V| = \{20, 30, 40, 50\}$. O conjunto de colunas $Inst.$ descreve as características das instâncias testadas, onde d representa a densidade do grafo. Cada linha descreve resultados médios de 10 instâncias que possuem os mesmos parâmetros $|V|$ e d. Os conjuntos de colunas EDGE, MLSteiner e SGMLSteiner representam os resultados de cada modelo. Estes conjuntos são divididos em: $\#$ que representa o número de instâncias resolvidas na otimalidade pelo modelo; $t(s)$ que indica o tempo computacional médio, em segundos, das 10 instâncias; lb e ub que representam, respectivamente, o valor médio do *lower bound* e *upper bound* obtido pelos modelos; $gapr$ indicando a porcentagem média da diferença entre a relaxação linear do modelo e seu lb; e, por fim, $nodes$ e $cuts$ denotam, respectivamente, a quantidade total de nós processados e de cortes dinâmicos aplicados durante o processo de *B&C*.

Podemos observar na Tabela 2 que as estratégias *MLSteiner* e *SGMLSteiner* obtiveram, respectivamente, 11 e 7 soluções ótimas a mais que o *EDGE*, bem como alcançaram melhores tempos computacionais médios em 10 de 12 grupos de instâncias. O *MLSteiner* obteve os melhores tempos médios em 50 instâncias de 120, com uma redução média de 13,2% do tempo nestas instâncias quando comparado ao *SGMLSteiner*, se mostrando a melhor estratégia para instâncias com $d = \{md, hd\}$. A estratégia *SGMLSteiner* obteve melhores tempos computacionais em outras 40 instâncias, reduzindo o tempo médio em apenas 10,6%, apresentando uma melhora em seu comportamento nas instâncias de baixa densidade e poucos vértices por causa do menor número de cortes a serem adicionados no grafo devido a diminuição do tamanho do conjunto de rótulos comparado a *MLSteiner* e *EDGE*. Por fim, a estratégia *SGMLSteiner* não conseguiu resolver na otimalidade uma das instâncias do grupo ($|V| = 40$, $d = ld$), atingindo o tempo limite de uma hora, observando sua execução a mesma obtém, em apenas 3 s de execução, o valor de *upper bound* igual 17, solução ótima da instância, e o valor 16 em seu *lower bound*, passando o resto da execução em suas ramificações do *branch-and-bound* resolvendo 479.568 nós sem conseguir fechar a solução final.

6. Considerações finais e proposta de trabalhos futuros

Neste trabalho, abordamos o Problema da Árvore de Cobertura com Representação Mínima (PACRM), neste, objetiva-se encontrar uma árvore de cobertura que minimize as cores representadas em cada vértice, tendo este problema aplicações na área de projetos de redes de computadores.

Propomos um modelo de programação linear inteira baseado em cortes coloridos para o Problema de Steiner de Rotulação Mínima, sendo assim, duas transformações polinomiais, intituladas MLSteiner e SGMLSteiner, de grafos-problema do PACRM foram apresentadas, de forma que a solução do modelo constrói uma solução para o PACRM. Estas duas estratégias foram comparadas, sendo o MLSteiner mais eficiente para instâncias com densidade de arestas $d = \{md, hd\}$. Já a estratégia SGMLSteiner foi mais eficiente para instâncias de menor densidade com $d = ld$.

Como planejamento de trabalhos futuros, pretendemos construir para métodos exatos uma estratégia de Geração de Colunas que utilize no modelo mestre o conceito de componente maximal monocromática conexa em suas colunas.

Tabela 2: Resultados dos métodos exatos para grafos com $|V| \in \{20, 30, 40, 50\}$

Inst.		EDGE						MLSteiner						SGMLSteiner					
$\|V\|$	d	#	t(s)	lb	ub	gapr	nodes	cuts	#	t(s)	lb	ub	gapr	nodes	cuts	#	t(s)	lb	ub	gapr	nodes	cuts
20	ld	10	0,065	12,3	12,3	2,1	165	60	10	0,061	12,3	12,3	2,1	254	40	10	**0,039**	12,3	12,3	2,1	**154**	**38**
	md	10	0,3	5,7	5,7	7,0	553	161	10	0,22	5,7	5,7	4,0	199	86	10	**0,17**	5,7	5,7	4,7	**148**	**85**
	hd	10	9,4	2,8	2,8	9,3	11094	375	10	**1,7**	2,8	2,8	5,8	**1428**	**206**	10	1,9	2,8	2,8	6,0	1641	215
30	ld	10	0,27	15,3	15,3	3,0	418	115	10	0,14	15,3	15,3	2,3	295	**64**	10	**0,11**	15,3	15,3	2,7	**255**	70
	md	10	10,3	6,7	6,7	5,7	6488	318	10	**3,1**	6,7	6,7	2,8	1661	**206**	10	3,5	6,7	6,7	3,1	**1617**	212
	hd	7	1712,9	3,0	4,0	12,3	366601	2677	**8**	**949,0**	**3,5**	**3,8**	7,6	120121	**2044**	**8**	1016,9	**3,5**	**3,8**	7,6	**106250**	2251
40	ld	**10**	**123,1**	**18,9**	18,9	3,9	**40860**	**620**	**10**	275,3	**18,9**	18,9	2,8	55209	736	9	361,8	18,8	18,9	2,8	51244	774
	md	7	1323,0	7,9	8,7	7.4	122967	3322	**9**	**593,5**	**8,3**	**8,5**	3,5	**33589**	**2068**	**9**	706,8	**8,3**	**8,5**	3,6	33878	2100
	hd	2	3256,9	1,8	6,6	14,3	304489	3353	**4**	2184,7	**3,3**	**4,7**	7,4	113512	**2685**	**4**	**2174,9**	**3,3**	4,8	7,6	**103777**	2919
50	ld	9	**109,8**	21,5	21,7	3,6	**61841**	380	**10**	213,5	**21,7**	21,7	2,2	90494	**255**	**10**	253,0	**21,7**	21,7	2,3	114679	332
	md	1	3250,8	8,1	11,3	9,7	255634	4992	**6**	**1723,6**	**9,3**	**9,8**	3,6	50725	4508	5	1873,5	**9,3**	9,9	3,7	**42811**	**4088**
	hd	0	3601,6	1,5	13,6	21,2	287141	**3936**	0	**3601,5**	**3,4**	**6,4**	7,9	120814	4377	0	**3601,5**	3,3	6,5	8,1	**106260**	4896

Referências

Barbosa, M. A. L. e Delbem, A. C. B. (2017). Uma busca local iterada para o problema da Árvore geradora mínima sob restrições de conflitos. *49º Simpósio Brasileiro de Pesquisa Operacional*.

Broersma, H. e Li, X. (1997). Spanning trees with many or few colors in edge-colored graphs. *Discussiones Mathematicae Graph Theory*, 17(2):259–269.

Cerrone, C., D'Ambrosio, C., e Raiconi, A. (2019). Heuristics for the strong generalized minimum label spanning tree problem. *Networks*, 74.

Cerulli, R., Fink, A., Gentili, M., e Voss, S. (2005). Metaheuristics comparison for the minimum labelling spanning tree problem. In Golden, B., Raghavan, S., e Wasil, E., editors, *The Next Wave in Computing, Optimization, and Decision Technologies*, volume 29 of *Operations Research/Computer Science Interfaces Series*, p. 93–106. Springer US. ISBN 978-0-387-23528-8. URL http://dx.doi.org/10.1007/0-387-23529-9_7.

Cerulli, R., Fink, A., Gentili, M., e Voss, S. (2006). Extensions of the minimum labelling spanning tree problem. *Journal of Telecommunications and Information Technology*.

Chen, C.-Y. e Hsieh, S.-Y. (2020). An efficient approximation algorithm for the steiner tree problem. In *Complexity and Approximation*, p. 238–251. Springer.

Chen, Y., Cornick, N., Hall, A. O., Shajpal, R., Silberholz, J., Yahav, I., e Golden, B. L. (2008). *Comparison of Heuristics for Solving the Gmlst Problem*, p. 191–217. Springer US, Boston, MA. ISBN 978-0-387-77780-1. URL https://doi.org/10.1007/978-0-387-77780-1_9.

Consoli, S., Darby-Dowman, K., Mladenovic, N., e Moreno-Pérez, J. (2009). Variable neighbourhood search for the minimum labelling steiner tree problem. *Annals of Operations Research*, 172:71–96.

Cormen, T. H., Leiserson, C. E., Rivest, R. L., e Stein, C. (2009). *Introduction to Algorithms, Third Edition*. The MIT Press, 3rd edition. ISBN 0262033844.

Gurguri, J. e Andrade, R. (2017). Modelos e desigualdades validas para o problema da Árvore-estrela. *49º Simpósio Brasileiro de Pesquisa Operacional*.

Linares, E. D. Z. (2018). Problema da Árvore geradora de representação mínima. Master's thesis, Universidade Federal Fluminense.

Linares, E. D. Z., Satoru, L., Silva, T. G., e Interian, R. (2018). The minimum representation spanning tree problem. *Anais do Simpósio Brasileiro de Pesquisa Operacional*, 274.

Silva, T. G. (2018). *The minimum labeling spanning tree and related problems*. PhD thesis, Universidade Federal Fluminense.

Silva, T. G. d., Queiroga, E., Ochi, L. S., Cabral, L., Gueye, S., e Michelon, P. (2018). A hybrid metaheuristic for the minimum labeling spanning tree problem. *European Journal of Operational Research*, 274.