# Novos Ótimos para o Poblema da Árvore Geradora com Rotulação Mínima

**Thiago Gouveia da Silva[1,3], Eduardo Vieira Queiroga[2], Luiz Satoru Ochi[3], Lucidio dos Anjos F. Cabral[2]**

[1]Instituto Federal de Educação, Ciência e Tecnologia da Paraíba(IFPB)

[2]Centro de Informática – Universidade Federal da Paraíba(UFPB)
João Pessoa – PB – Brasil

[3]Instituto de Computação – Universidade Federal Fluminense(UFF)
Niterói – RJ – Brasil

thiago.gouveia@ifpb.edu.br, eduardovqueiroga@gmail.com
satoru@ic.uff.br, lucidio@ci.ufpb.br

## RESUMO

O Problema da Árvore Geradora com Rotulação Mínima (PAGRM) consiste em: dado um grafo não orientado $G = (V, E, L)$, sendo $V$ o conjunto de vértices, $E$ o conjunto de arestas e $L$ o conjunto de rótulos, no qual cada aresta $e \in E$ possui um conjunto de rótulos $L(e)$ associado; o objetivo é encontrar uma árvore geradora de $G$ que utilize o menor número de rótulos possível. O PAGRM possui aplicações no projeto de redes de comunicação homogêneas assim como na geração de redes de transporte multimodais. Este trabalho realiza um estudo comparativo entre a Formulação Baseada em Cortes Coloridos e o melhor método exato da literatura, propõe uma nova estratégia de *branch-and-bound*, um novo conjunto de inequações válidas e duas heurísticas de separação. Os experimentos realizados demonstram que os métodos propostos apresentaram excelente desempenho, sendo capazes de provar todos os ótimos, dos quais doze inéditos, para o grupo de instâncias de Cerulli *et al*. (2005) com $|V| = 200$.

**PALAVRAS CHAVE. Estratégias de Branching. Planos de corte. Árvore Geradora**

**Área Principal: Programação Matemática**

## ABSTRACT

The Coloring Spanning Tree Problem (CSTP) consists in: given a non-directed graph $G = (V, E, L)$, where $V$ is the set of vertices, $E$ is the set of edges and $L$ is the set of labels. Every edge $e \in E$ has a set of associated labels $L(e)$. The goal is to find a spanning tree of $G$ that uses a minimum number of labels. The CSTP has applications on homogeneous comunication networks design and on multimodal transportation networks generation. This work performs a comparative study between the FBCC mathematical formulation and the best exact method, presents a new branch-and-bound strategy, a new set of valid inequalities and two separation heuristics. The experiments show that the proposed methods have competitive performance, proving all optimum, twelve all-new, for the $|V| = 200$ instance set of Cerulli *et al*. (2005).

**KEYWORDS. Branching Strategies. Cutting planes. Spanning Tree**

**Main Area: Mathematical Programming**

## 1. Introdução

O problema da Árvore Geradora com Rotulação Mínima (PAGRM), proposto por Chang e Leu (1997), consiste em: dado um grafo não orientado no qual cada aresta possui um conjunto de rótulos (cores) associado, encontrar uma árvore geradora que utilize o menor número possível de rótulos (cores). O PAGRM pertence à classe NP-Difícil e possui aplicações em redes de comunicação (Consoli *et al.*, 2009), redes de transporte multimodais (Van-Nes, 2002) e compressão de dados (Chwatal *et al.*, 2009).

Consoli *et al.* (2009) descreve a aplicação do PAGRM para o projeto de redes de comunicação nas quais existem diferentes meios de transporte possíveis entre os nós da rede, por exemplo cabos de fibra ótica mono/multimodo, de par trançado, coaxiais ou enlaces com diferentes tecnologias de comunicação sem fio. O objetivo seria projetar uma rede possibilitando a conectividade entre todos seus nós utilizando a menor quantidade de meios de comunicação, proporcionando, assim, uma rede mais homogênea.

Formalmente, o PAGRM pode ser descrito como segue: dado um grafo não orientado $G = (V, E, L)$, sendo $V$ o conjunto de vértices de $G$, $E$ o conjunto de arestas e $L$ o conjunto de rótulos (cores) sobre $E$, onde cada aresta $e \in E$ possui um conjunto de rótulos $L(e)$ associado; o objetivo é encontrar uma árvore $T = (V, E', L')$, tal que $E' \subseteq E$ e $L' \subseteq L$ e $|L'|$ seja minimizado, ou seja, que utilize o menor número de rótulos possível na árvore solução.

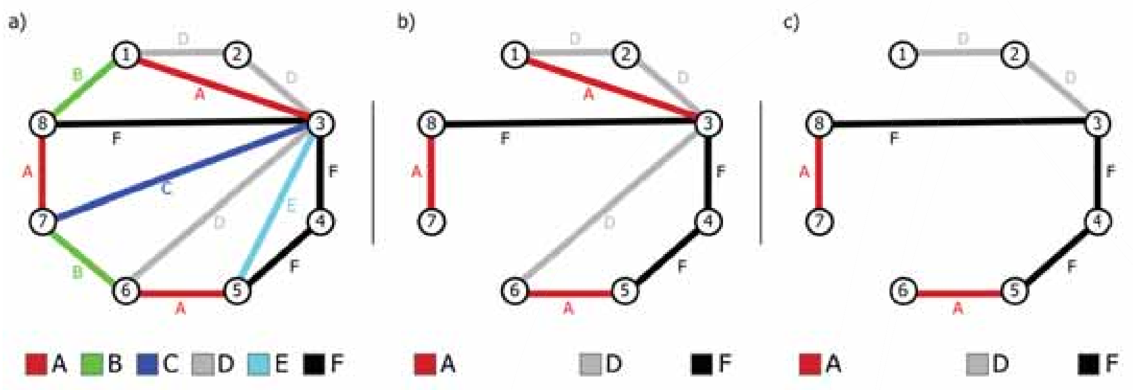


**Figura 1. (a) Instância-exemplo do PAGRM, (b) Grafo Conexo com Rotulação Mínima, (c) Árvore de Geradora com Rotulação Mínima.**

A Figura 1(a) expõe uma pequena instância-exemplo (PIE) para o PAGRM, para qual $V = \{1, 2, 3, 4, 5, 6\}$, $L = \{A, B, C, D, E, F\}$ e cada aresta $e$ possui apenas um rótulo associado, indicado pela letra próxima a esta. A Figura 1(b) exibe um grafo conexo com rotulação mínima para a PIE, enquanto a Figura 1(c) apresenta uma solução para esta PIE, formada pelo subgrafo $T$ acíclico e conexo, rotulado pelo conjunto $L' = \{A, D, F\}$, cujo custo é $|L'| = 3$, valor ótimo para a instância.

Vale ressaltar que, apesar de definirmos uma árvore como um grafo conexo e acíclico, o PAGRM pode ser resolvido sem levar em consideração a proibição de ciclos. Conforme observado por Xiong *et al.* (2005), seja $C^* \subseteq L$ o menor conjunto de rótulos tal que o subgrafo $G^*$ induzido por $C^*$ seja conexo, qualquer árvore de cobertura $T^*$ de $G^*$ possuirá no máximo $|C^*|$ rótulos. Dado $G^*$, uma árvore de cobertura $T^*$ de $G^*$ pode ser obtida em tempo polinomial.

Neste trabalho é apresentado um estudo comparativo entre os métodos exatos existentes para resolução do PAGRM, uma nova estratégia de *branch-and-bound*, um novo conjunto de inequações válidas para o problema e novas instâncias com resultados ótimos provados. Para tal, a Seção 2 traz uma breve revisão sobre os trabalhos relacionados ao PAGRM com foco nos métodos exatos; a Seção 3 apresenta a Formulação Baseada em Cortes Coloridos (FBCC) e as técnicas propostas para seu aprimoramento; A Seção 4 analisa os experimentos e resultados obtidos e, por fim, as considerações finais e propostas de trabalhos futuros são discutidos na Seção 5.

## 2. Trabalhos Relacionados

Desde que foi proposto, o PAGRM tem sido abordado por diversas heurísticas, meta-heurísticas e métodos exatos. Chang e Leu (1997) propuseram dois algoritmos heurísticos para o PAGRM, dos quais destaca-se o Algoritmo da Cobertura Máxima de Vértices (MVCA, do inglês *Maximum Vertex Covering Algorithm*), sendo utilizado por algumas meta-heurísticas para (re)construir soluções. O MVCA parte do grafo sem arestas e, a cada iteração, adiciona ao grafo todas as arestas com o rótulo que cobre o maior número de vértices ainda não visitados. O procedimento termina quando todos os vértices tiverem sido cobertos. Krumke e Wirth (1998) propuseram uma versão corrigida do MVCA, dado que este poderia produzir soluções inviáveis.

No mesmo trabalho, Krumke e Wirth (1998) provam que o MVCA possui o fator de aproximação $1 + 2log(n)$, para $n = |V|$. Tal limite foi aprimorado para $ln(n-1)+1$ por Wan *et al*. (2002), enquanto Xiong *et al*. (2006) provam que uma solução do MVCA apresenta fator de aproximação igual ao *b-ésimo* número harmônico, para um grafo com frequência máxima dos rótulos limitada por $b$.

A primeira meta-heurística proposta para o PAGRM foi o Algoritmo Genético (GA, do inglês *Genetic Algorithm*) de Xiong *et al*. (2005), obtendo resultados superiores aos do MVCA. Cerulli *et al*. (2005) analisaram o desempenho das meta-heurísticas Busca Tabu Reativa, Simulated Annealing e VNS, enquanto Xiong *et al*. (2006) publicaram uma aprimoração do GA, denominada MGA (do inglês *Modified Genetic Algorithm*), alcançando os melhores resultados para o problema até então. Consoli *et al*. (2009) implementou métodos baseados nas meta-heurísticas GRASP e VNS, superando o MGA. Chwatal e Raidl (2010) desenvolveu um algoritmo de Colônia de Formigas para o problema.

Recentemente, Silva *et al*. (2014) propuseram o algoritmo híbrido MSLB (do inglês, *Multi Start Local Branching*) para o PAGRM, baseado em uma implementação eficiente do MVCA em conjunto com uma nova estrutura de vizinhança que faz uso de técnicas de *Local Branching* (Fischetti e Lodi, 2003). Consoli *et al*. (2015), por sua vez, desenvolveram duas novas abordagens: COMPL e INTELL, que utilizam conceitos de Inteligência Artificial e Aprendizagem de Máquina para aprimorar o VNS. Os métodos INTELL e MSLB possuem os melhores resultados heurísticos para o problema até o presente momento.

Em Chang e Leu (1997) foi proposta a primeira abordagem exata para o PAGRM, baseada no algoritmo de busca em árvores $A^*$. O método consiste em criar uma árvore de rótulos explorados que possui uma estimativa de custo para o valor da solução final, guiando a construção e a exploração da árvore até que seja encontrado um nó objetivo, que é indexado por rótulos que produzem uma solução ótima. Captivo *et al*. (2009) desenvolveram três formulações matemáticas baseadas em fluxo simples (SCF, do inglês *Single Commodity Flow*) para resolver o PAGRM por MIP (*Mixed Integer Programming*).

Em Chwatal e Raidl (2011) é realizado um estudo extensivo sobre técnicas de programação matemática para a resolução do PAGRM no qual são propostas cinco novas formulações, planos de corte e heurísticas de separação para os conjuntos exponenciais de restrições, estratégias de *branch-and-cut*, *branch-and-cut-and-price* e diversos conjuntos de inequações válidas para melhorar a relaxação linear dos modelos.

A primeira formulação estudada por Chwatal e Raidl (2011) é a SFC, que foi reformulada dando origem a um novo modelo, baseado em múltiplos fluxos (MCF, do inglês *Multi Commodity Flow*). A segunda formulação proposta é baseada em cortes de arcos (DCUT, do inglês *Directed Cut*), possuindo número exponencial de restrições, separadas por uma heurística de fluxo máximo. Em seguida são propostos dois novos modelos baseados em eliminação de ciclos: CEF (do inglês, *Cycle-Elimination Formulation*) e MTZ (formulação de Miller-Tucker-Zemlin). Por fim, é apresentada uma formulação baseada em cortes de arestas (EC, do inglês *Epsilon Connectivity*), possuindo o número exponencial de restrições separadas por um algoritmo de busca em profundidade em grafos.

Os melhores resultados exatos para o PAGRM até o momento foram obtidos pela formulação EC$_{sn}$, dada pelo formulação EC fortalecida por inequações de ligação forte (índice s, do inglês *Strong Linkage*) e de rotulação de nós (índice n, do inglês *Node Label Constraints*). Uma vez que os métodos propostos neste trabalho tem o desempenho comparado com a formulação EC$_{sn}$, esta é discutida em mais detalhes na Seção 2.1.

### 2.1. Formulação EC$_{sn}$

A formulação EC$_{sn}$, proposta por Chwatal e Raidl (2011) define os seguintes conjuntos de variáveis: as variáveis de rótulos (cores) $z_l \in \{0, 1\}$, para todo $l \in L$ indicam se o rótulo $l$ é utilizado na solução; as variáveis de aresta $x_e$, para todo $e \in E$, indicam se a aresta $e$ é usada na árvore geradora da solução final. A formulação EC$_{sn}$ é descrita em (1)..(9).

A função objetivo (1) minimiza o número de rótulos (cores) utilizados na solução. O conjunto de inequações (2) garante que para cada aresta *e* escolhida, pelo menos um de seus rótulos é selecionado. A restrição (3) garante o número válido de arestas ativadas para formar uma árvore geradora.

Seja $\epsilon \in \mathbb{R}$, um número arbitrariamente pequeno, digamos $0,0001$; seja $\delta(S)$ o conjunto de arestas incidentes ao subconjunto $S \subset V$; o conjunto exponencial de restrições do tipo *conectividade epsilon* (4) obriga que para todo $S$ exista pelo menos uma aresta incidente ativa, garantindo a conectividade do grafo solução. Este conjunto de restrições é inicialmente vazio, sendo adicionado ao modelo à medida que são encontradas violações. Dada uma solução para a relaxação linear do EC, a heurística de separação escolhe arbitrariamente um vértice a partir do qual é executado um algoritmo de busca em profundidade de grafos (DFS) considerando apenas as arestas $e$ com $x_e >= \epsilon$. Caso o DFS não consiga alcançar todos os vértices do grafo, um corte válido foi encontrado.

O conjunto de inequações (5) garante que exista pelo menos uma aresta ativa incidente a cada vértice $v \in V$, enquanto as inequações (6) e (7) definem os domínios e limites da variáveis $x$ e $z$, respectivamente. Dado que o conjunto de variáveis $z \in \{0,1\}^{|L|}$, as restrições (2), que associam as variáveis de rótulos às variáveis de arestas, possibilitam que as variáveis do conjunto $x$ sejam contínuas.

$$\text{Minimizar} \sum_{l \in L} z_l \tag{1}$$

$$\begin{aligned}
\text{s.t.} \quad \sum_{l \in L(e)} z_l &\geq x_e, & \forall e \in E, \quad (2)\\
\sum_{e \in E} x_e &= |V| - 1, & \quad (3)\\
\sum_{e \in \delta(S)} x_e &>= \epsilon, & \forall S \subset V, S \neq \emptyset, \quad (4)\\
\sum_{e \in \delta(v)} x_e &>= 1, & \forall v \in V, \quad (5)\\
x_e &\geq 0, & \forall e \in E, \quad (6)\\
z_l &\in \{0,1\}, & \forall l \in L. \quad (7)
\end{aligned}$$

Dois grupos de restrições são adicionadas ao EC para tornar a sua relaxação linear mais forte, dando origem à formulação EC$_{sn}$: as inequações de ligação forte (índice s, do inglês *Strong Linkage*) (8) associam as variáveis de aresta diretamente às variáveis de rótulos, substituindo o conjunto de inequações (2) e excluindo as restrições (3), enquanto o conjunto de restrições de rotulação de nós (índice n, do inglês *Node Label Constraints*) (9) garante que exista pelo menos um rótulo ativo incidente a cada vértice $v \in V$.

$$\begin{aligned}
z_{L(e)} &= x_e, & \forall e \in E, \quad (8)\\
\sum_{l \in L(\delta(v))} z_l &>= 1, & \forall v \in V. \quad (9)
\end{aligned}$$

## 3. Formulação Baseada em Cortes Coloridos (FBCC)

A Formulação Baseada em Cortes Coloridos (FBCC), proposta por Silva *et al.* (2014), foi utilizada em conjunto com técnicas de *Local Branching* para propiciar uma estratégia de busca local exata para o PAGRM. A meta-heurística resultante, MSLB, obteve resultados competitivos em relação aos melhores algoritmos heurísticos da literatura. No entanto, não foi realizado nenhum estudo comparativo contemplando a FBCC e os melhores métodos exatos. Neste trabalho é analisado o desempenho da FBCC em relação ao EC$_{sn}$, assim como são propostas duas melhorias para a FBCC: uma nova estratégia de *branch-and-bound* e um novo conjunto de inequações válidas, discutidas nas Seções 3.1 e 3.2, respectivamente.

A FBCC garante a conectividade do grafo solução obrigando que todo corte de arestas do grafo possua pelo menos um rótulo ativo, o que implica que exista uma ou mais arestas ativas para todo corte. Formalmente, seja $\delta(S)$ o conjunto de arestas do corte $[S, V\backslash S]$ e $L(\delta(S))$ o conjunto de rótulos associados às arestas de $\delta(S)$, para todo subconjunto $S \subset V$, $S \neq \emptyset$, a solução deve conter pelo menos um rótulo $l \in L(\delta(S))$ ativo, e por consequência uma ou mais arestas conectando $S$ ao conjunto de vértices $V\backslash S$. A formulação FBCC é apresentada nas equações (10) a (12).

$$\text{Minimizar} \sum_{l \in L} z_l \tag{10}$$

$$\text{s.t.} \quad \sum_{l \in L(\delta(S))} z_l \geq 1, \qquad \forall S \subset V, V \neq \emptyset \tag{11}$$

$$z_l \in \{0, 1\}, \qquad \forall l \in L. \tag{12}$$

A FBCC define apenas o grupo de variáveis $z_l \in \{0, 1\}$, $\forall l \in L$, que indica se o rótulo $l$ faz parte da solução, enquanto a função objetivo (10) minimiza o número de rótulos utilizados. O conjunto de restrições (11) garante a conectividade do grafo solução resultante e o conjunto de restrições (12) define o domínio das variáveis $z_l$.

Pode-se observar que a FBCC apresenta um conjunto exponencial de restrições (11), tornando impraticável a resolução do modelo completo para instâncias de médio e grande porte. Por este motivo, o conjunto de inequações (11) é substituído inicialmente pelo conjunto (13), que garante a conectividade do grafo para todo conjunto unitário de vértices (*singletons*) de $G$.

$$\sum_{l \in L(\delta(v))} z_l \geq 1, \qquad \forall v \in V. \tag{13}$$

As restrições não consideradas inicialmente são geradas sob demanda por um *framework* de planos de corte. O algoritmo utilizado para separação das inequações (11) efetua uma busca em profundidade no grafo induzido pela solução da FBCC e, para cada componente conexa encontrada, é adicionada uma nova restrição ao modelo. O procedimento é repetido até que o grafo solução seja conexo.

### 3.1. Estratégia de *Branching* Prévio

A resolução de problemas formulados como MIPs geralmente faz uso de algoritmos de *branch-and-bound* que consistem em particionar o espaço de soluções em subespaços disjuntos e solucionar os submodelos resultantes. Após a resolução da relaxação linear (RL) do modelo é selecionada uma variável $x$ com valor fracionário a partir da qual são gerados dois subproblemas: um para $x \leq \lfloor \overline{x} \rfloor$ e outro para $x \geq \lceil \overline{x} \rceil$ ($x = 0$ e $x = 1$ para variáveis binárias).

A estrutura do conjunto de restrições (13) da FBCC, contudo, possibilita que seja utilizada uma nova estratégia de particionamento, denominada *branching* por vértices (BpV). Dado um vértice $v \in V$ qualquer, pelo menos uma variável $z_l$ para $l \in L(\delta(v))$ deve ser 1; então, para cada $l \in L(\delta(v))$ é definido um novo subproblema no qual $z_l = 1$.

Para algumas instâncias, porém, o BpV poder ser mais agressivo (BpVa) ao combinar dois vértices que não possuam nenhum rótulo em comum: sejam $v, w \in V$ tal que $L(\delta(v)) \cap L(\delta(w)) = \emptyset$, pelo menos um par de variáveis $z_l, l \in L(\delta(v))$ e $z_k, k \in L(\delta(w))$ deve possuir valor $1$. Neste caso, é criado um subproblema no qual cada par de variáveis $z_l = z_k = 1$. Não é possível aplicar o BpVa quando o grafo-problema não possui nenhum par de vértices sem rótulos em comum.

A estratégia de *branching* prévio proposta realiza o BpVa (BpV) sobre a FBCC, entregando cada subproblema resultante para que o resolvedor aplique o algoritmo de *branch-and-bound* tradicional. A solução do modelo completo é dada pela melhor solução dos subproblemas. A Figura 2(a) apresenta uma instância do PAGRM na qual os vértices $1$ e $2$ não possuem rótulos em comum. A Figura 2(b) demonstra o *branching* prévio: o segundo nível da árvore é composto pelos subproblemas gerados pelo BpVa onde cada caixa exibe os rótulos fixados em 1, enquanto os níveis seguintes exibem a resolução do subproblema $z_A = z_E = 1$ pelo método tradicional. A Figura 2(c) exibe o grafo resultante após a fixação dos rótulos $A$ e $C$ enquanto a Figura 2(d) detalha o grafo resultante após a colapsação dos vértices em respeito aos rótulos fixados.

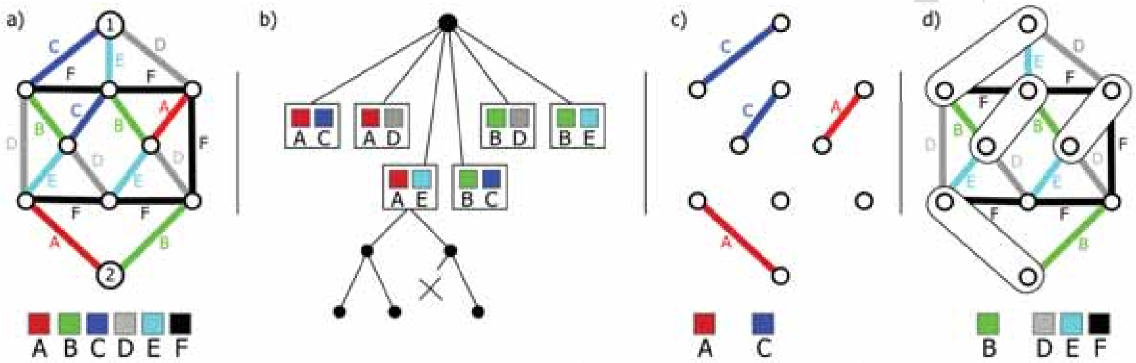


**Figura 2. Ilustração da execução do *Branching* Prévio para uma instância do PAGRM.**

### 3.2. Cortes de Cores Incidentes

Com o objetivo de fortalecer a FBCC com respeito a sua relaxação linear e, possivelmente, diminuir o tempo de execução do algoritmo de *branch-and-cut* utilizado para resolução do modelo, propomos um novo conjunto de inequações válidas para o PAGRM, denominado Cortes de Cores Incidentes (IC). Os cortes IC tem como base o critério de que devem existir arestas suficientes para que qualquer subconjunto de vértices seja conexo entre si e se conecte ao restante do grafo.

Dado um conjunto de vértices $S \subset V$, a inequação (11) relacionada garante que o corte $\delta(S)$ possua pelo menos uma aresta ativa; neste caso, o conjunto $S$ precisa de pelo menos $|S| - 1$ arestas internas ativas para ser conexo. Cada aresta ativa adicional no corte $\delta(S)$ diminui em $1$ o limite inferior de arestas internas ativas. Podemos observar, então, que a soma das arestas internas ativas com as arestas ativas do corte $\delta(S)$ deve ser pelo menos $|S|$. Formalmente, seja $Ne(l, S)$ uma função que retorna o número de arestas com rótulo $l$ que incidem sobre qualquer vértice $v \in S$, o conjunto de inequações (14) define o conjunto exponencial de cortes IC.

$$\sum_{l \in L} Ne(l,S)z_l \geq |S|, \qquad \forall S \subset V, S \neq \emptyset \qquad (14)$$

Seja $G^p$ um grafo-problema, a Figura 3(a) exibe a solução da relaxação linear da FBCC inicializada apenas com restrições de *singletons*. A Figura 3(b) explicita que o conjunto $S = \{1,2,3,4,5,7,8\}$ viola o corte de IC associado e exibe o corte adicionado ao modelo, enquanto a Figura 3(c) demonstra a solução da relaxação linear da FBCC após a adição do corte de IC. Para $G^p$, em especial, o novo conjunto de restrições melhorou a relaxação linear da FBCC de $2.5$ para $3.5$, além de possibilitar a obtenção de uma solução com três variáveis inteiras.

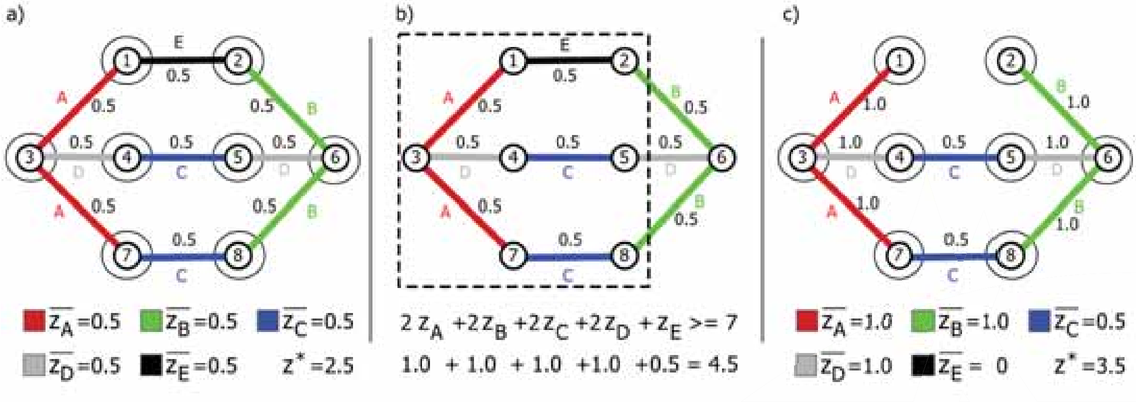


**Figura 3. (a) Relaxação linear do modelo FBCC em singletons, (b) Corte encontrado pela heurística de separação para o IC, (c) Relaxação linear após adição do corte.**

A Heurística de Corte Mais Violado por Raiz (HCMV$_r$) é o procedimento utilizado como base para as heurísticas de separação propostas. A HCMV$_r$ parte de um vértice raiz e aumenta o conjunto $S$ de forma gulosa em busca de cortes de IC violados, utilizando a função $b(v, \overline{z}, S)$, que retorna a soma dos valores da relaxação linear das arestas com um extremo em $v$ e outro em $S$. As funções $cutSum(S)$ e $intSum(S)$ retornam a soma das arestas do corte $\delta(S)$ e a soma das arestas com dois extremos em $S$, respectivamente.

**Algoritmo 1:** Heurística de Corte Mais Violado por Raiz- HCMV$_r$

```
1  Procedimento HCMV_r (root)
2      S ← {root};
3      Cut ← ∅;
4      maiorViolacao ← −∞;
5      enquanto |S| < |V| faça
6          S ← S ∪ {minback(V\S)};
7          violacao ← |S| − cutSum(S) − intSum(S);
8          se violacao > 0 E violacao > maiorViolacao então
9              Cut ← S;
10             maiorViolacao ← violacao;
11     retorne Cut;
```

O Algoritmo 1 descreve a (HCMV$_r$). As linhas 2-4 inicializam as estruturas de dados; as linhas 5-10 trazem o laço principal do método; na linha 6 é adicionado ao conjunto $S$ o vértice com o valor $minback(V \backslash S) = min\{b(v_k) \forall v_k \in V \backslash S)\}$; a linha 7 calcula a violação para o conjunto $S$; enquanto as linhas 8-10 mantém o melhor corte, que é retornado pelo procedimento na linha 11.

A primeira heurística de separação proposta: Heurística de Corte Mais Violado (HCMV), executa a HCMV$_r$ para toda raiz possível e retorna o melhor corte encontrado. A segunda heurística de separação proposta: HCMV Múltipla (HCMVm) é uma simples modificação na HCMV para que esta retorne o primeiro corte violado encontrado para cada tamanho diferente de $S$. Ambas heurísticas de separação são aplicadas apenas no nó raiz da árvore de *branch-and-bound* da FBCC ou na raiz de cada subproblema gerado pelo *branching* prévio.

## 4. Resultados Computacionais

Todos os métodos estudados neste trabalho foram desenvolvidos em linguagem C++, utilizando o compilador g++ versão 4.6.3 (opção de compilação -O3), com o auxilio do resolvedor matemático CPLEX versão 12.5.1. Os experimentos computacionais foram executados em um computador com processador *Intel Core* I7 (64 bits), com CPU contendo 4 *cores* de 3.4 GHz, além de 16 GB de RAM e executando o sistema operacional Linux Ubuntu 14.04 LTS. Os experimentos foram realizados com $4$ *threads* e tempo limite de duas horas, sendo expandido para quatro horas para o experimento detalhado.

As instâncias usadas nos experimentos possuem 120 grafos-problema com $|V| = 200$ gerados de forma aleatória por Cerulli *et al*. (2005). O conjunto consiste em $12$ grupos de instâncias com distintas dimensões, sendo cada grupo definido por uma combinação entre $|L| = \{50; 100; 200; 250\}$ e $d = \{ld = 0,2; md = 0,5; hd = 0,8\}$, $d$ representando a densidade do grafo. Cada grupo possui $10$ instâncias distintas de mesma dimensão.

A Tabela 1 apresenta o resultado do primeiro experimento realizado, que teve como objetivo o estudo comparativo do desempenho da FBCC em relação ao EC$_{sn}$, melhor método exato da literatura. Cada linha da tabela está relacionada com um grupo de $10$ instâncias de mesma dimensão, descrito no primeiro conjunto de colunas. A coluna *Med* indica a média dos valores ótimos obtidos para o grupo de instâncias. As colunas restantes são divididas em dois conjuntos: *EC$_{sn}$* e *FBCC*, que possuem as colunas *O*, indicando o número de instâncias resolvidas, *t(s)* que representa o tempo total (em segundos) que o método levou para resolver todas as instâncias, *Min t* (*Max t*) que exibe o menor (maior) tempo necessário para resolver uma única instância e *B*, reportando o número de vezes que o método obteve o melhor tempo de execução. De modo adicional, o grupo *FBCC* possui a coluna $\Delta t$ que representa a porcentagem da diferença (*gap*) entre os tempos totais de execução dos dois métodos.

Podemos observar na Tabela 1 que a FBCC obteve desempenho superior ao EC$_{sn}$, alcançando menores tempos de execução em 11 de 12 grupos de instâncias, além de apresentar um *gap* expressivo de $-51\%$ para o grupo de instâncias mais difíceis: 200-ld-250. Além disto, o FBCC conseguiu os melhores tempos para todas as instâncias individuais em 4 grupos, vencendo por 9 a 1 em outros 4.

O segundo experimento, reportado na Tabela 2, foi realizado sobre os três conjuntos de instâncias mais difíceis (200-md-250, 200-ld-200 e 200-ld-250) com objetivo de analisar o desempenho das novas estratégias propostas e detalhar os resultados de forma individual para cada instância. Uma vez que a FBCC apresentou desempenho superior, os

**Tabela 1. Resultados computacionais dos métodos exatos para instâncias com $|V| = 200$.**

| *Instâncias* | | | *EC$_{sn}$* | | | | | *FBCC* | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $\|L\|$ | *d* | *Med* | *O* | *t (s)* | *Min t* | *Max t* | *B* | *O* | *t (s)* | *$\Delta$ t* | *Min t* | *Max t* | *B* |
| 250 | ld | 13,8 | 7 | 15057,4 | 240,99 | 6490,0 | 1 | 7 | **7407,25** | **-51%** | 25,35 | 2853,3 | 6 |
| | md | 6,3 | 10 | **1335,4** | 38,39 | 529,48 | 6 | 10 | 1477,49 | 11% | 12,85 | 334,80 | 4 |
| | hd | 4,0 | 10 | 89,81 | 1,71 | 12,15 | 0 | 10 | **49,11** | **-45%** | 1,08 | 9,40 | **10** |
| 200 | ld | 11,9 | 9 | 4842,57 | 40,19 | 2594,7 | 1 | 10 | **2037,29** | **-58%** | 19,15 | 6919,1 | **9** |
| | md | 5,4 | 10 | 325,00 | 5,51 | 143,56 | 2 | 10 | **268,18** | **-17%** | 2,56 | 96,55 | 8 |
| | hd | 4,0 | 10 | 89,78 | 1,71 | 12,15 | 0 | 10 | **49,11** | **-45%** | 1,07 | 9,35 | **10** |
| 100 | ld | 7,9 | 10 | 123,60 | 1,46 | 39,05 | 1 | 10 | **48,84** | **-60%** | 0,46 | 15,79 | **9** |
| | md | 3,4 | 10 | 8,15 | 0,42 | 1,75 | 2 | 10 | **6,27** | **-23%** | 0,146 | 2,88 | 8 |
| | hd | 2,6 | 10 | 3,14 | 0,21 | 0,39 | 1 | 10 | **2,12** | **-32%** | 0,078 | 0,40 | **9** |
| 50 | ld | 5,2 | 10 | 3,85 | 0,15 | 0,69 | 1 | 10 | **2,23** | **-42%** | 0,06 | 0,52 | **9** |
| | md | 2,2 | 10 | 1,24 | 0,10 | 0,15 | 0 | 10 | **0,35** | **-71%** | 0,02 | 0,06 | **10** |
| | hd | 2,0 | 10 | 1,55 | 0,15 | 0,16 | 0 | 10 | **0,18** | **-88%** | 0,01 | 0,02 | **10** |

aprimoramentos foram aplicados sobre esta: o índice $p$ indica a utilização do *branching* prévio (BP), enquanto os índices $b$ e $m$ indicam o uso dos cortes de cores incidentes (IC) com as heurísticas de separação HCMV e mHCMV, respectivamente. Cada linha da tabela representa uma instância; a primeira coluna, *# Inst.*, traz o número da instância no grupo; a coluna *Ót.* reporta o valor ótimo; a coluna *Nós*$_p$ exibe o número de nós gerados pelo BP enquanto as colunas restantes apresentam os tempos que cada método levou para obter o resultado. Células com *NF* indicam que o método não concluiu a execução dentro do limite de tempo.

Podemos observar pela Tabela 2 que tanto o BP quanto as heurísticas de separação de IC conseguem melhorar o desempenho da FBCC em várias instâncias, tendo a FBCC$_p$ obtido o melhor desempenho para o grupo 200-ld-200. A combinação do BP com os cortes de IC separados pela heurística HCMV, contudo, produz resultados ainda mais expressivos: A FBCC$_{pb}$ apresenta o menor tempo de execução total para os grupos 200-md-250 e 200-ld-250, alcançando o ótimo para todas as instâncias exceto uma, que foi resolvida pela FBCC$_b$. Vale ressaltar que os métodos explorados neste trabalho conseguiram provar todos os ótimos para o grupo de instâncias de Cerulli *et al*. (2005) com $|V| = 200$, dentre estes, quatro novos ótimos para instâncias com $|L| = 200$ e oito para $|L| = 250$.

## 5. Considerações Finais e Proposta de Trabalhos Futuros

Neste trabalho é abordado o Problema de Árvore Geradora de Rótulação Mínima (PAGRM), no qual dado um grafo não orientado $G = (V, E, L)$, onde cada aresta $e \in E$ possui um conjunto de rótulos $L(e)$ associado; o objetivo é encontrar uma árvore geradora de $G$ que utilize o menor número de rótulos possível. Foi realizado um estudo comparativo entre a FBCC e o melhor método exato da literatura; foi proposta uma nova estratégia de *branch-and-bound*, denominada *branching* prévio (BP); um novo conjunto de inequações válidas (cortes de cores incidentes, IC) e duas heurísticas de separação.

Os experimentos computacionais realizados demonstraram que a FBCC obteve desempenho superior ao EC$_{sn}$, alcançando melhores tempos de execução para todos os grupos de instâncias, exceto o 200-md-250. Também pudemos observar que tanto o BP

**Tabela 2. Resultados computacionais detalhados para instâncias difíceis.**

***Instâncias com*** $|V| = 200$***,*** $|L| = 250$ ***e d=md***

| *# Inst.* | *Ót.* | *FBCC* | $\mathit{EC}_{sn}$ | $\mathit{FBCC}_b$ | $\mathit{FBCC}_m$ | $\mathit{FBCC}_p$ | $\mathit{FBCC}_{pb}$ | $\mathit{FBCC}_{pm}$ | $\mathit{Nós}_p$ |
|---|---|---|---|---|---|---|---|---|---|
| 0 | 6 | 172,43 | 60,56 | 45,34 | 257,99 | 67,80 | **44,71** | 63,22 | 60 |
| 1 | 7 | 12,85 | 38,39 | **12,18** | 14,29 | 40,75 | 59,96 | 72,31 | 48 |
| 2 | 6 | 249,32 | 126,31 | **33,57** | 55,27 | 183,64 | 43,89 | 65,94 | 63 |
| 3 | 7 | 113,18 | 529,48 | 166,91 | 67,18 | **26,54** | 48,29 | 52,82 | 45 |
| 4 | 6 | **31,57** | 110,69 | 52,61 | 93,16 | 117,57 | 48,93 | 71,18 | 65 |
| 5 | 6 | 158,21 | 139,97 | 118,53 | 223,17 | 344,26 | **48,99** | 68,80 | 65 |
| 6 | 7 | 45,03 | 49,49 | 47,90 | 59,21 | **39,90** | 57,26 | 71,96 | 41 |
| 7 | 6 | 125,18 | 61,93 | 175,15 | 238,59 | 65,23 | **42,86** | 61,91 | 58 |
| 8 | 6 | 234,91 | 127,59 | 143,03 | 72,97 | 292,15 | **44,59** | 64,23 | 60 |
| 9 | 6 | 334,80 | 91,00 | 115,29 | **35,49** | 79,59 | 44,62 | 64,27 | 60 |
| ***Total*** | 63 | 1477,49 | 1335,40 | 910,51 | 1117,32 | 1257,43 | **484,09** | 656,62 | 565 |

***Instâncias com*** $|V| = 200$***,*** $|L| = 200$ ***e d=ld***

| *# Inst.* | *Ót.* | *FBCC* | $\mathit{EC}_{sn}$ | $\mathit{FBCC}_b$ | $\mathit{FBCC}_m$ | $\mathit{FBCC}_p$ | $\mathit{FBCC}_{pb}$ | $\mathit{FBCC}_{pm}$ | $\mathit{Nós}_p$ |
|---|---|---|---|---|---|---|---|---|---|
| 0 | 13 | **107,02** | 262,10 | 351,54 | 313,21 | 350,45 | 466,18 | 448,21 | 240 |
| 1 | 13 | 43,05 | 431,32 | **39,16** | 39,17 | 148,31 | 267,09 | 266,53 | 288 |
| 2 | 10 | 6919,11 | 13000,30 | 7578,18 | 4587,48 | 302,16 | **241,54** | 263,43 | 589 |
| 3 | 13 | 405,57 | **138,99** | 439,91 | 441,23 | 522,45 | 658,75 | 656,78 | 323 |
| 4 | 12 | 100,37 | 159,44 | 129,37 | **54,04** | 61,66 | 160,94 | 161,86 | 240 |
| 5 | 10 | 664,65 | 1015,40 | 654,71 | 315,84 | 480,29 | **193,74** | 215,72 | 456 |
| 6 | 14 | **586,96** | 2594,68 | NF | 2855,04 | 1637,166 | 1777,08 | 1776,83 | 180 |
| 7 | 12 | 91,22 | 117,18 | **70,11** | 256,24 | 156,83 | 343,02 | 414,36 | 255 |
| 8 | 11 | **19,15** | 83,28 | 36,83 | 67,82 | 74,45 | 128,75 | 172,51 | 225 |
| 9 | 11 | **19,31** | 40,19 | 157,32 | 82,28 | 72,83 | 221,15 | 258,53 | 288 |
| ***Total*** | 119 | 8956,40 | 17842,87 | 9457,13 | 9012,35 | **3806,609** | 4458,23 | 4634,76 | 3084 |

***Instâncias com*** $|V| = 200$***,*** $|L| = 250$ ***e d=ld***

| *# Inst.* | *Ót.* | *FBCC* | $\mathit{EC}_{sn}$ | $\mathit{FBCC}_b$ | $\mathit{FBCC}_m$ | $\mathit{FBCC}_p$ | $\mathit{FBCC}_{pb}$ | $\mathit{FBCC}_{pm}$ | $\mathit{Nós}_p$ |
|---|---|---|---|---|---|---|---|---|---|
| 0 | 14 | **25,35** | 1019,78 | 98,74 | 29,15 | 789,25 | 1044,71 | 1103,46 | 240 |
| 1 | 16 | NF | NF | **12476,54** | NF | 14337,26 | NF | NF | 289 |
| 2 | 12 | 11778,37 | 12787,70 | NF | NF | 9032,87 | **445,34** | 672,03 | 513 |
| 3 | 15 | 1514,73 | 4334,85 | **459,57** | 1697,75 | 873,74 | 982,93 | 956,03 | 306 |
| 4 | 14 | **60,87** | 241,00 | 179,12 | 171,86 | 205,93 | 432,81 | 463,46 | 240 |
| 5 | 12 | NF | NF | NF | NF | NF | **987,99** | 1281,00 | 680 |
| 6 | 15 | 2574,64 | 6490,00 | 2601,57 | 2596,05 | **1663,02** | 1713,05 | 1713,88 | 180 |
| 7 | 14 | 2853,33 | 2227,73 | 3851,92 | 5049,71 | 2093,95 | **1921,88** | 3102,68 | 240 |
| 8 | 12 | 80,47 | 331,17 | 40,87 | **31,73** | 80,21 | 187,53 | 258,39 | 240 |
| 9 | 13 | 297,87 | 412,91 | **199,00** | 558,10 | 531,96 | 246,56 | 387,19 | 288 |
| ***Total*** | 137 | 19185,63 | 27854,12 | 19907,33 | 10134,35 | 29608,18 | **7262,79** | 9938,14 | 3216 |

quanto os cortes IC são capazes de melhorar o desempenho da FBCC, conseguindo provar doze novos ótimos para o PAGRM, fechando, assim, o grupo de instâncias de Cerulli *et al.* (2005) com $|V| = 200$.

Para a continuidade da pesquisa propomos a exploração de meta-heurísticas e métodos exatos, especificamente modelos matemáticos, para separação de cortes de cores incidentes com objetivo de fortalecer a relaxação linear da FBCC. Um outro caminho a ser seguido é a proposição de novos modelos para o PAGRM, possibilitando a sua resolução por técnicas de geração de colunas e *branch-and-cut-and-price*.

## Referências

**Captivo, M., Clímaco, J. a. C. N. e Pascoal, M. M. B.** (2009), A mixed integer linear formulation for the minimum label spanning tree problem. *Comput. Oper. Res.*, v. 36, n. 11, p. 3082–3085.

**Cerulli, R., Fink, A. e Gentili, Monica e Voß, S.** Metaheuristics comparison for the minimum labelling spanning tree problem. Golden, B., Raghavan, S. e Wasil, E. (Eds.), *The Next Wave in Computing, Optimization, and Decision Technologies*, volume 29 of *Operations Research/Computer Science Interfaces Series*, p. 93–106. Springer US, 2005.

**Chang, R.-S. e Leu, S.-J.** (1997), The minimum labeling spanning trees. *Inf. Process. Lett.*, v. 63, n. 5, p. 277–282.

**Chwatal, A. M. e Raidl, G. R.** Solving the minimum label spanning tree problem by ant colony optimization. Arabnia, H. R., Hashemi, R. R. e Solo, A. M. G. (Eds.), *GEM*, p. 91–97. CSREA Press. ISBN 1-60132-145-7, 2010.

**Chwatal, A. M. e Raidl, G. R.** (2011), Solving the minimum label spanning tree problem by mathematical programming techniques. *Adv. Operations Research*, v. 2011.

**Chwatal, A. M., Raidl, G. R. e Oberlechner, K.** (2009), Solving a *k*-node minimum label spanning arborescence problem to compress fingerprint templates. *J. Math. Model. Algorithms*, v. 8, n. 3, p. 293–334.

**Consoli, S., Mladenović, N. e Moreno Pérez, J.** (2015), Solving the minimum labelling spanning tree problem by intelligent optimization. *Appl. Soft Comput.*, v. 28, n. C, p. 440–452.

**Consoli, S., Darby-Dowman, K., Mladenovic, N. e Moreno-Pérez, J. A.** (2009), Greedy randomized adaptive search and variable neighbourhood search for the minimum labelling spanning tree problem. *European Journal of Operational Research*, v. 196, n. 2, p. 440 – 449.

**Fischetti, M. e Lodi, A.** (2003), Local branching. *Mathematical Programming*, v. 98, n. 1-3, p. 23–47.

**Krumke, S. O. e Wirth, H.-C.** (1998), On the minimum label spanning tree problem. *Inf. Process. Lett.*, v. 66, n. 2, p. 81–85.

**Silva, T. G., Queiroga, E. V., de Sousa Filho, G. F., Cabral, L. A. F. e Ochi, L. S.** Abordagem híbrida para o problema da Árvore geradora com rotulacão mínima. *Anais do XLVI Simpósio Brasileiro de Pesquisa Operacional*, p. 1823–1833, 2014.

**Van-Nes, R.** *Design of Multimodal Transport Networks: A Hierachical Approach*. Tese de doutorado, Delft University, 2002.

**Wan, Y., Chert, G. e Xu, Y.** (2002), A note on the minimum label spanning tree. *Inf. Process. Lett.*, v. 84, n. 2, p. 99–101.

**Xiong, Y., Golden, B. e Wasil, E.** (2006), Improved heuristics for the minimum label spanning tree problem. *Evolutionary Computation, IEEE Transactions on*, v. 10, n. 6, p. 700–703.

**Xiong, Y., Golden, B. L. e Wasil, E. A.** (2005), A one-parameter genetic algorithm for the minimum labeling spanning tree problem. *IEEE Trans. Evolutionary Computation*, v. 9, n. 1, p. 55–60.