

# Abordagem Híbrida para o Problema da Árvore Geradora com Rotulação Mínima

Thiago Gouveia da Silva<sup>1,3</sup>, Eduardo Vieira Queiroga<sup>2</sup>, Gilberto F. de Sousa Filho<sup>2,3</sup>,  
Lucidio dos Anjos F. Cabral<sup>2</sup>, Luiz Satoru Ochi<sup>3</sup>

<sup>1</sup>Instituto Federal de Educação, Ciência e Tecnologia da Paraíba(IFPB)  
João Pessoa – PB – Brasil

<sup>2</sup>Centro de Informática – Universidade Federal da Paraíba(UFPB)  
João Pessoa – PB – Brasil

<sup>3</sup>Instituto de Computação – Universidade Federal Fluminense(UFF)  
Niteroi – RJ – Brasil

thiago.gouveia@ifpb.edu.br, {gilberto, lucidio}@ci.ufpb.br, satoru@ic.uff.br

## RESUMO

O Problema da Árvore Geradora de Rotulação Mínima (PAGRM) consiste em: dado um grafo não orientado  $G = (V, E, L)$ , sendo  $L$  o conjunto de rótulos,  $E$  o conjunto de Arestas,  $L'$  e  $E'$  seus respectivos subconjuntos; encontrar uma árvore  $T = (V, E', L')$  em  $G$ , tal que  $|L'|$  seja minimizado. O PAGRM possui aplicações no projeto de redes de comunicação homogêneas assim como na geração de redes de transporte multimodais. Este trabalho apresenta uma nova formulação matemática com número linear de variáveis (FBCC) para o problema; um *framework* de planos de corte para a FBCC; uma nova estrutura de vizinhança exata baseada no conceito de *Local Branching* ( $LB_k$ ); e uma abordagem híbrida (MSLB) que combina características das meta-heurísticas *Multi Start* e VND com a  $LB_k$ . Por fim, é realizado um estudo comparativo que evidencia que os resultados obtidos pela MSLB se mostraram bastante competitivos em comparação com a literatura da área.

**PALAVRAS CHAVE.** Meta-heurística híbrida. Planos de corte. Local branching

**Área Principal:** Meta-heurística, Programação Matemática

## ABSTRACT

The Minimum Labeling Spanning Tree Problem (MLSTP) consists in: given a non-directed graph  $G = (V, E, L)$ , where  $L$  is the set of labels (colors) over  $E$ ,  $L'$  and  $E'$  are the respective subsets; find one tree  $T = (V, E', L')$  of  $G$  that minimizes  $|L'|$ . The MLSTP has applications on homogeneous communication networks design and on multimodal transportation networks generation. This work presents a new mathematical formulation with linear number of variables (FBCC) for the problem; a new cutting-planes framework to solve the FBCC; a new Local Branching based exact neighborhood structure ( $LB_k$ ); and a hybrid approach (MSLB) that blends the Multi Start and VND metaheuristics characteristics with the  $LB_k$  proposed. Finally, a comparative study was done evidencing that the MSLB results are competitive in respect to the MLSTP literature ones.

**KEYWORDS.** Hybrid metaheuristics. Cutting planes. Local branching

**Main Area:** Metaheuristic, Mathematical Programming

## 1. Introdução

O problema da Árvore Geradora Mínima (PAGM) é um problema clássico em Teoria dos Grafos que consiste em encontrar uma árvore (subgrafo acíclico e conexo) de um grafo que possui arestas ponderadas positivamente, conectando todos os vértices e minimizando a soma dos custos das arestas do grafo resultante. Existem vários algoritmos clássicos que resolvem esse problema em tempo polinomial, dentre eles os mais conhecidos são os algoritmos clássicos de Kruskal e Prim.

Apesar do problema PAGM ser resolvido em tempo polinomial, algumas de suas variantes são problemas de otimização combinatória classificados com complexidade NP-Completo e adequadas para abordagens aproximativas, tais como heurísticas e meta-heurísticas. A variante abordada neste trabalho é o problema da Árvore Geradora de Rotulação Mínima (PAGRM), proposto por Chang e Leu (1997), que também provaram que o problema é NP-Completo. No PAGRM, dado um grafo não orientado  $G = (V, E, L)$ ,  $V$  representa o conjunto de vértices (ou nós) do grafo,  $E$  representa o conjunto de arestas e  $L$  o conjunto de rótulos (ou cores) sobre  $E$ , onde cada aresta  $e \in E$  possui um conjunto de rótulos  $L(e)$  associado. O objetivo é encontrar uma árvore  $T = (V, E', L')$  em  $G$ , cujo  $|L'|$  seja minimizado, ou seja, que utilize o número de rótulos utilizados na árvore solução.

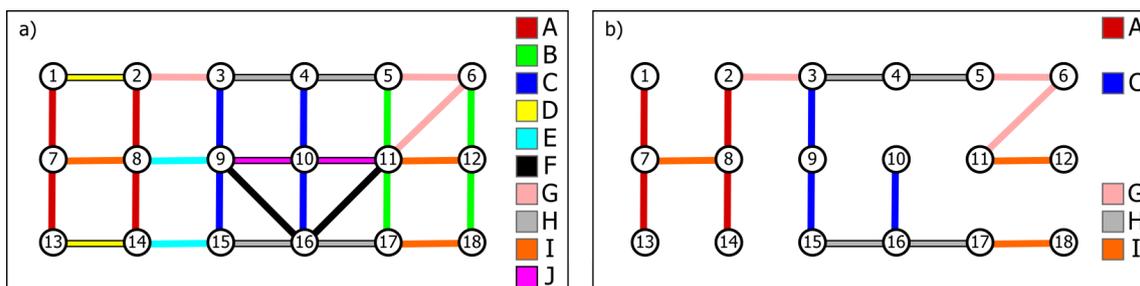


Figura 1. (a) Instância-exemplo do problema. (b) Solução ótima da instância.

Podemos observar na Figura 1(a) uma instância para o PAGRM, onde  $V = \{1..18\}$  e  $L = \{A..J\}$ . Cada aresta  $e$  possui apenas um rótulo para facilitar o entendimento da instância. A Figura 1(b) apresenta uma solução para esta instância, formado pelo subgrafo  $T$  acíclico e conexo, rotulado pelo conjunto  $L' = \{A, C, G, H, I\}$ , cujo custo é igual a 5 que é o valor ótimo para a instância.

Segundo Consoli *et al.* (2009), no projeto de redes de comunicação onde existem vários meios de comunicação entre os nós da rede (p.e. fibra ótica, par trançado, rede sem fio ou cabo coaxial), encontrar uma árvore que conecte todos os nós da rede com a quantidade mínima de tipos de meio de comunicação é uma aplicação real do PAGRM.

Um outro exemplo prático é dado pelas redes de transporte multimodais VanNes (2002). Neste problema, é necessário prover um serviço completo usando o número mínimo de companhias. Uma rede de transporte multimodal é representada por um grafo onde para cada aresta é atribuída um rótulo, representando uma companhia que gerencia o serviço naquela aresta. O objetivo é encontrar uma árvore geradora usando o número mínimo de companhias.

Este problema já foi abordado na literatura com técnicas exatas e heurísticas.

Neste contexto, propomos um método híbrido para o PAGRM, adotando em sua fase de construção um procedimento *MultiStart* e em sua busca local o uso de um *Local Branching*. Para isto, na seção 2 descrevemos a formulação matemática do PAGRM e seus trabalhos anteriores. Na seção 3 descrevemos uma nova formulação matemática para o PAGRM além de um plano de cortes para esta formulação. Descrevemos ainda o método híbrido com sua fase de construção de uma solução viável e o *Local Branching* proposto (seção 4). Os Resultados computacionais são apresentados na seção 5 e por fim as considerações finais na seção 6.

## 2. Problema da Árvore Geradora com Rotulação Mínima

### 2.1. Formulação *Directed Cut*

Chwatal e Raidl (2011) definem uma formulação para o PAGRM, chamada formulação *Directed Cut*, nela são definidas as seguinte variáveis:  $z_l \in \{0, 1\}$ , para todo  $l \in L$  indica se o rótulo  $l$  é parte da solução; variáveis de aresta  $x_e$ , para todo  $e \in E$ , indica se a aresta  $e$  é usada na solução da árvore geradora final; variáveis  $y_{i,j}$ , para todo  $i, j \in V$ , indica a existência de um arco direcionado para cada aresta  $e = \{i, j\} \in E$  podendo indicar a existência do arco  $(i, j)$  ou  $(j, i) \in A$ . Seja  $L(e)$  o conjunto de rótulos associados à aresta  $e$ . A formulação *Directed Cut* proposta é apresentada a seguir.

$$\text{Minimizar } \sum_{l \in L} z_l \quad (1)$$

$$\text{s.t. } \sum_{l \in L(e)} z_l \geq x_e, \quad \forall e \in E, \quad (2)$$

$$\sum_{e \in E} x_e = |V| - 1, \quad (3)$$

$$\sum_{(i,j) \in \delta^-(S)} y_{ij} \geq 1, \quad \forall S \subset V, 0 \notin S, \quad (4)$$

$$x_e \geq y_{ij}, \quad \forall \{i, j\} \in E, e = \{i, j\}, \quad (5)$$

$$z_l \in \{0, 1\}, \quad \forall l \in L. \quad (6)$$

A função objetivo (1) minimiza o número de rótulos 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  $\delta^-(S)$  o conjunto de arcos incidentes a algum subconjunto  $S \subset V$ , o conjunto de restrições (4) garante que cada subconjunto  $S$  tenha pelo menos um arco incidente ativo sobre ele, garantindo a conectividade do grafo solução. O conjunto de inequações (5) associa os arcos direcionados aos rótulos.

Os conjuntos de inequações (7) e (8) podem ser adicionados para obter vantagens computacionais em formulações direcionadas baseadas em cortes (Chimani *et al.*, 2010).

$$y_{ij} + y_{ji} \leq 1, \quad \forall \{i, j\} \in E, \quad (7)$$

$$\sum_{(i,j) \in \delta^-(j)} y_{ij} \geq 1, \quad j \in V \setminus \{0\}. \quad (8)$$

O Conjunto de inequações (7) proíbe ciclos curtos de uma mesma aresta. O conjunto de inequações (8) assegura que cada vértice tenha pelo menos um arco incidente. As inequações (8) devem ser adicionadas quando for realizado um plano de cortes sobre as inequações (4). Por  $\delta^-(i)$  consideramos o conjunto de arcos incidentes ao vértice  $i$ .

## 2.2. Trabalhos Relacionados

Em Chang e Leu (1997) foi proposto um algoritmo exato para o problema PAGRM baseado no algoritmo  $A^*$ , uma técnica de busca em árvore. Este algoritmo cria 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.

Já em Chwatal e Raidl (2011) foram propostas 5 formulações matemáticas para o PAGRM, planos de cortes para as restrições exponenciais e os métodos exatos *Branch & Cut* e *Branch&Cut&Price* que encontraram o ótimo de novas instâncias de pequeno porte. Dentre as formulações apresentadas, os melhores resultados foram obtidos pela *Epsilon Cut* e *Directed Cut*, como descrito na seção 2.1.

O primeiro algoritmo heurístico para o PAGRM foi proposto em Chang e Leu (1997), chamado de Algoritmo da Cobertura Máxima de Vértices (*MVCA - Maximum Vertex Covering Algorithm*). Nesta heurística, a partir de um grafo vazio, são adicionadas as arestas com o rótulo que possui maior cobertura de vértices ainda não visitados, escolhendo novos rótulos enquanto o grafo não for conectado. Em caso de empate, um dos rótulos entre o conjunto selecionado é escolhido aleatoriamente. Krumke e Wirth Krumke e Wirth (1998) propuseram uma versão corrigida do *MVCA*, que é apresentado no Algoritmo 1, alegando que o original poderia gerar soluções não conectadas. Também provaram que o *MVCA* corrigido constrói soluções com fator de aproximação  $(1 + 2 \log n)$ , onde  $n$  é o número de vértices do grafo. Esse limite foi melhorado por Wan et al., Wan et al. (2002), que provaram que o *MVCA* corrigido possui um fator de aproximação  $(\ln(n - 1) + 1)$ , sendo  $n > 1$ . Já em Xiong et al. (2006) foi provado que a solução do *MVCA*, para um grafo com frequência máxima dos rótulos igual a  $b$ , possui fator de aproximação igual ao  $b$ -ésimo número harmônico.

Duas meta-heurísticas foram propostas em Consoli et al. (2009) para o problema: GRASP (Greedy Reandomized Adaptive Search Procedure) e VNS (Variable Neighbourhood Search). Essas duas abordagens se mostraram mais eficientes que o Método Piloto proposto em Xiong et al. (2006) e o MGA (Modified Genetic Algorithm) proposto em Cerulli et al. (2005), sendo este VNS o método mais eficiente da literatura para instâncias de grande dimensão.

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

Esta seção descreve a Formulação Baseada em Cortes Coloridos (FBCC), proposta para o PAGRM, baseada na formulação *Directed Cut* (*DCUT*) descrita na seção 2.1. A ideia

---

**Algoritmo 1: Algoritmo MVCA corrigido**

---

```

1 procedimento MVCA ( $G = (V, E, L)$ )
2    $f(s^o) \leftarrow \infty$ ;
3   Seja  $C \leftarrow \emptyset$  o conjunto de rótulos utilizados;
4   Seja  $H = (V, E', C)$  um subgrafo de  $G$  com as arestas que possuem rótulo em  $C$ ;
5   Seja  $numComp(H)$  o número de componentes conectadas de  $H$ ;
6   enquanto  $numComp(H) > 1$  faça
7     Selecione o rótulo  $c \in (L \setminus C)$  que minimiza  $numComp(H)$ ;
8     Atualize o conjunto  $C \leftarrow C \cup \{c\}$  e o grafo  $H = (V, E', C)$ ;
9   Encontre uma árvore qualquer  $T$  de  $H = (V, E', C)$ ;
```

---

central da FBCC é que para qualquer subconjunto  $S \subset V$ ,  $S \neq \emptyset$ , deve haver pelo menos uma aresta no corte de arestas  $\delta(S)$  (e por consequência um rótulo) conectando  $S$  ao grupo de vértices  $V \setminus S$ . Segue o modelo:

$$\text{Minimizar } \sum_{l \in L} z_l \quad (9)$$

$$\text{s.t. } \sum_{l \in L(\delta(S))} z_l \geq 1, \quad \forall S \subset V, V \neq \emptyset \quad (10)$$

$$z_l \in \{0, 1\}, \quad \forall l \in L. \quad (11)$$

A FBCC define apenas a variável  $z_l \in \{0, 1\}$ ,  $\forall l \in L$ , que indica se o rótulo  $l$  faz parte da solução. A função objetivo (9) minimiza o número de rótulos utilizados. Seja  $L(\delta(S))$  o conjunto de rótulos associados ao corte de arestas  $\delta(S)$ , o conjunto de restrições (10) obriga para todo subconjunto  $S$  a existência de pelo menos um rótulo selecionado  $l \in L(\delta(S))$ , garantindo, assim, a conectividade do grafo solução resultante.

A FBCC pode ser considerada uma versão simplificada do DCUT, uma vez que utiliza a mesma ideia de cortes de arestas para garantir a conectividade do grafo resultante. No entanto, a não utilização de variáveis de decisão para arcos (variável  $y$ ) e arestas (variável  $x$ ) permite a substituição dos conjuntos de restrições (2), (3), (4) e (5) pelo conjunto (10).

É fácil observar que a FBCC apresenta um conjunto exponencial de restrições (10), assim a resolução do modelo com todas as restrições torna-se impraticável para instâncias de médio e grande porte. Por este motivo, foi proposto um *framework* de planos de corte para a resolução da FBCC (FPC\_FBCC), apresentado no Algoritmo 2. O FPC\_FBCC consiste em executar a FBCC para as inequações associadas às componentes conexas  $\mathcal{S}$  do subgrafo  $H$ , inicialmente sem arestas. O algoritmo de separação das inequações proposto é o Busca em Profundidade, (DFS, do inglês *Depth First Search*). Os rótulos escolhidos na execução da FBCC irão induzir novas arestas em  $H$ , adicionando à  $\mathcal{S}$  as novas componentes conexas de  $H$ . Este processo deve ser repetido até que  $H$  se torne um grafo conexo.

A Figura 2 ilustra a execução do FPC\_FBCC para uma pequena instância exemplo do PAGRM. Em a), é apresentada a instância e o grupo inicial de componentes conexas

**Algoritmo 2:** Algoritmo Plano de Corte para a FBCC

```

1 procedimento FPC_FBCC ( $G = (V, E, L)$ )
2   Seja  $C \leftarrow \emptyset$  o conjunto de rótulos utilizados;
3   Seja  $H = (V, E', C)$  um subgrafo de  $G$  com as arestas que possuem rótulo em  $C$ ;
4   Seja  $\mathcal{S} \leftarrow DFS(H)$  o algoritmo de separação que retorna os subconjuntos  $S \subset V$ 
   conexos em  $H$ ;
5   enquanto  $numComp(H) > 1$  faça
6      $C \leftarrow FBCC(\mathcal{S})$ ;
7     Atualize  $H = (V, E', C)$ ;
8      $\mathcal{S} \leftarrow \mathcal{S} \cup DFS(H)$ ;
9   retorne  $H$ ;
    
```

que será considerado pela FBCC:  $\mathcal{S} = \{\{1\}, \{2\}, \dots, \{8\}\}$ . Em b) podemos verificar a solução do FBCC para o primeiro grupo  $\mathcal{S}$ . Em c) podemos observar que para cada componente conexa do grafo de b) foi gerado um novo conjunto de vértices e adicionado à  $\mathcal{S}$ :  $\mathcal{S} = \mathcal{S} \cup \{\{1, 2, 3, 4\}, \{5, 6, 7, 8\}\}$ . Em d) podemos observar o resultado da execução da FBCC para o novo grupo  $\mathcal{S}$ . Como o grafo resultante é conexo, a FBCC chegou a uma solução ótima para a instância.

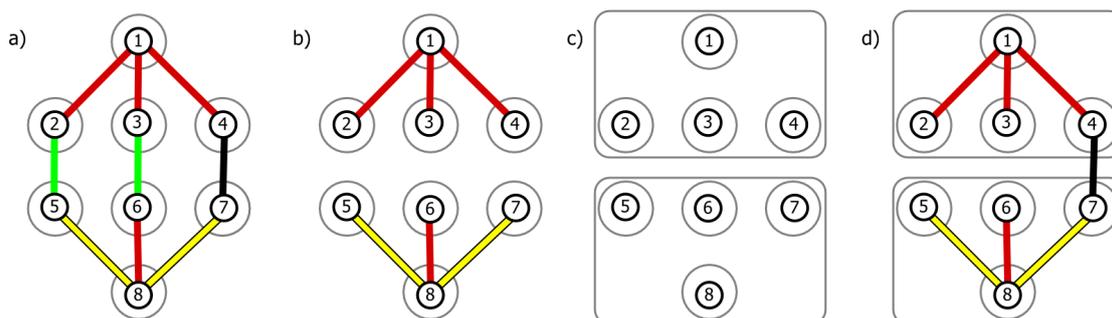


Figura 2. Ilustração da execução do algoritmo FPC\_FBCC para uma instância do PAGRM.

#### 4. Abordagem Híbrida para o PAGRM

Os diversos métodos propostos para o PAGRM até então podem ser divididos em duas classes: métodos heurísticos (como as meta-heurísticas MGA, VNS e GRASP), e métodos exatos (como as formulações matemáticas DCUT e FBCC). Os métodos exatos são capazes de alcançar a solução ótima em tempo aceitável para pequenas instâncias. As meta-heurísticas, em contrapartida, trocam a garantia de otimalidade por um menor tempo de execução e apresentam resultados satisfatórios para médias e grandes instâncias.

Nos últimos anos, contudo, evidenciou-se que a concentração apenas em um método apresenta-se bastante restritivo. A combinação de vários métodos heurísticos (meta-heurísticas) ou a combinação de métodos heurísticos e exatos (denominados meta-heurísticas híbridas) pode proporcionar um comportamento mais eficiente, como podemos observar nos trabalhos (Subramanian *et al.*, 2012) e (Subramanian *et al.*, 2013).

Esta seção apresenta uma abordagem híbrida para o PAGRM, doravante referenciada como MSLB, que combina a diversificação propiciada pela meta-heurística MultiStart (MS) com a intensificação da meta-heurística VND, utilizando uma nova estrutura de

vizinhança que realiza um *Local Branching* (LB) usando o método exato FPC\_FBCC. O algoritmo 3 apresenta a visão geral do MSLB. As subseções seguintes discutem o procedimento MS\_MVCA usado como heurística de construção, a nova estrutura de vizinhança baseada em LB proposta, e o framework VND utilizado como heurística de busca local.

---

**Algoritmo 3:** Algoritmo Híbrido para o PAGRM

---

```

1 procedimento MSLB ( $G = (V, E, L)$ )
2    $f(s^\circ) \leftarrow \infty$ ;
3   enquanto  $tempo < tempo\_limite$  faça
4      $s \leftarrow MS\_MVCA(G)$  //heurística construtiva ;
5      $s \leftarrow VND(s, G, \{LB_2, LB_3\})$  //procedimento de busca local ;
6      $s^\circ \leftarrow melhor(s^\circ, s)$ ;
7   Retorne uma árvore qualquer  $T$  de  $H = (V, E', C)$ ;
```

---

#### 4.1. Heurística de construção MS\_MVCA

A fase de construção do MSLB utiliza uma versão modificada da heurística MVCA apresentada na seção 2.2 envolta em um laço MultiStart (MS) que retorna a melhor solução encontrada em todas as iterações. As duas modificações no MVCA foram baseadas na adaptação da meta-heurística GRASP para o PAGRM proposta por Consoli *et al.* (2009):

- O primeiro rótulo é adicionado aleatoriamente ao grupo C de rótulos escolhidos;
- Em caso de empate durante a seleção do próximo rótulo que será adicionado à solução, a escolha é feita aleatoriamente.

As duas modificações efetuadas no MVCA permitem que uma área maior do espaço de soluções seja visitada, gerando, assim, soluções mais diversas, enquanto a estrutura MS desconsidera soluções de baixa qualidade. Uma vez que o objetivo do MS\_MVCA não é produzir uma solução final, mas entregar soluções de boa qualidade de diferentes partes do espaço de soluções, ele não executa nenhuma busca local. Esta característica o diferencia do GRASP.

Uma implementação eficiente do MVCA modificado tem papel importante na execução do MSLB. Esta possibilita um número grande de execuções do MS num tempo relativamente pequeno, possibilitando assim a produção de soluções diversificadas e de boa qualidade.

#### 4.2. Nova estrutura de vizinhança proposta

Proposto por Fischetti e Lodi (2003), o *Local Branching* (LB) é uma estratégia genérica para resolução de problemas inteiros mistos. O princípio do LB é realizar uma execução rápida do *solver* sobre uma solução de referência  $s$ , mesmo que a solução  $s'$  gerada seja apenas superficialmente melhor que  $s$ . A execução rápida do LB é alcançada pela adição de uma nova restrição ao modelo original que proíba a geração de soluções muito diferentes de  $s$ .

Dada uma solução de referência  $s$  para o PAGRM, a estratégia de LB adotada (formalizada pela adição da restrição (12) à FBCC) garante que a nova solução  $s'$  gerada pelo FPC\_FBCC utilize exatamente  $|s| - k$  rótulos pertencentes a  $s$ , onde  $k$  representa um parâmetro de proximidade entre as soluções  $s$  e  $s'$ . Para valores pequenos de  $k$ , o

FPC\_FBCC deve encontrar soluções muito parecidas com  $s$ , o que restringe o número de soluções possíveis e reduz bastante o tempo de execução do modelo.

$$\sum_{l \in s} z_l = |s| - k \quad (12)$$

Neste trabalho definimos uma nova estrutura de vizinhança parametrizável baseada no LB para o PAGRM. Para cada  $k \in \{1, 2, \dots, n\}$ , existe uma vizinhança  $LB_k$  que pode ser explorada como procedimento de busca local para uma dada solução. Uma característica importante da estrutura de vizinhança proposta é que  $\forall k, w \in \{1, 2, \dots, n\}$ , onde  $k \neq w$ ,  $LB_k \cap LB_w = \emptyset$ , isto é, não há sobreposição entre quaisquer vizinhanças da estrutura  $LB_k$ .

### 4.3. Variable Neighborhood Descent (VND)

Como método de busca local, foi adotado o *framework* Variable Neighborhood Descent (VND), apresentado em Hansen *et al.* (2008), com a aplicação do conjunto de vizinhanças  $N = \{LB_2, LB_3\}$  na ordem em que aparecem no grupo.

Em experimentos iniciais sobre as vizinhanças  $LB_k$  foram analisados os valores de  $k$  entre 1 e 6. Para  $k \in \{4, 5, 6\}$  percebeu-se um grande aumento no tempo de execução do  $LB_k$ , tornando-o pouco atraente para o problema, enquanto a vizinhança  $LB_1$ , apesar de rápida, não apresentou resultados que justificassem a sua utilização. Sendo este o motivo da adoção de  $LB_2$  e  $LB_3$  em nossa abordagem híbrida.

## 5. Resultados Computacionais

A abordagem híbrida MSLB proposta neste trabalho foi desenvolvida na linguagem C++, utilizando o compilador g++ versão 4.6.3 (opção de compilação -O3), com o auxílio do solver matemático CPLEX versão 12.5.1. Os experimentos computacionais foram executados em uma máquina 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 12.04 LTS.

O conjunto de instâncias utilizadas nos experimentos possui 240 problemas gerados por Cerulli *et al.* (2005). Este conjunto consiste de grupos de instâncias com distintas dimensões, sendo cada grupo é definido pela combinação das seguintes dimensões  $|V| = \{200; 500\}$ ,  $|L| = \{\frac{|V|}{4}; \frac{|V|}{2}; |V|; 5\frac{|V|}{4}\}$ , e a densidade  $d$  das arestas do grafo:  $d = \{ld = 0, 2; md = 0, 5; hd = 0, 8\}$ , além disso, cada grupo possui 10 instâncias aleatórias distintas de mesma dimensão.

Para a execução do MSLB utilizamos como condição de parada o tempo computacional, sendo 60 segundos para as instâncias com  $|V| = 200$  e 300 segundos para  $|V| = 500$ , estes valores foram os mesmos adotados na literatura comparada. Executamos o MSLB 10 vezes para cada uma das 240 instâncias. Em cada execução foram realizadas iterações do algoritmo até que o tempo limite fosse alcançado, sendo cada iteração composta por 500 construções do procedimento MS\_MVCA e uma execução do VND sobre a melhor solução da fase de construção, sempre registrando a melhor solução encontrada em todas as iterações.

Instância			MSLB				VNS		GRASP		Pilot		MGA	
$ V $	$ L $	$d$	$Z_{min}$	$\Delta(\%)$	$Z_{med}$	$T$	$Z$	$T$	$Z$	$T$	$Z$	$T$	$Z$	$T$
200	250	hd	<b>4</b>	0	<b>4</b>	0,35	<b>4</b>	1,1	4,1	4,9	<b>4</b>	9,1	<b>4</b>	2,2
		md	<b>6,3</b>	0	<b>6,3</b>	0,32	<b>6,3</b>	3,4	<b>6,3</b>	0,51	<b>6,3</b>	8,4	<b>6,3</b>	17,6
		ld	<b>13,9</b>	0	<b>13,9</b>	0,86	<b>13,9</b>	3,2	<b>13,9</b>	1,4	<b>13,9</b>	8	14	26,4
200	200	hd	<b>4</b>	0	<b>4</b>	0,35	<b>4</b>	0,03	<b>4</b>	0,04	<b>4</b>	5,9	<b>4</b>	0,2
		md	<b>5,4</b>	0	<b>5,4</b>	0,28	<b>5,4</b>	0,97	<b>5,4</b>	0,89	5,5	5,6	<b>5,4</b>	16,1
		ld	<b>12</b>	0	12,11	4,01	<b>12</b>	12,8	12,2	9,4	12,4	5	12,4	12,7
200	100	hd	<b>2,6</b>	0	<b>2,6</b>	0,27	<b>2,6</b>	0,12	<b>2,6</b>	0,05	<b>2,6</b>	0,88	<b>2,6</b>	0,14
		md	<b>3,4</b>	0	<b>3,4</b>	0,21	<b>3,4</b>	0,15	<b>3,4</b>	0,17	<b>3,4</b>	1,2	<b>3,4</b>	1,6
		ld	8	1,26	8,1	1,47	<b>7,9</b>	1,7	8,1	0,67	8,3	1,3	8,3	2,2
200	50	hd	<b>2</b>	0	<b>2</b>	0,18	<b>2</b>	0	<b>2</b>	0,02	<b>2</b>	0,09	<b>2</b>	0,03
		md	<b>2,2</b>	0	<b>2,2</b>	0,14	<b>2,2</b>	0,02	<b>2,2</b>	0,01	<b>2,2</b>	0,16	<b>2,2</b>	0,07
		ld	<b>5,2</b>	0	<b>5,2</b>	0,08	<b>5,2</b>	0,24	<b>5,2</b>	0,04	<b>5,2</b>	0,32	<b>5,2</b>	0,32

**Tabela 1. Comparação dos resultados computacionais dos algoritmos aproximativos para instâncias com  $|V| = 200$ .**

As Tabelas 1 e 2 apresentam a comparação dos resultados computacionais da abordagem híbrida MSLB proposta com 4 algoritmos aproximativos da literatura. Cada linha das Tabelas 1 e 2 está relacionada com um grupo de 10 instâncias de mesma dimensão. Nestas tabelas temos o primeiro grupo de colunas representando os grupos de instâncias testados e as colunas restantes são divididas em cinco grupos *MSLB*, *VNS*, *GRASP*, *Pilot* e *MGA*. O grupo de colunas *MSLB* possui a coluna  $Z_{min}$  indicando a média dos valores das melhores soluções encontradas;  $\Delta(\%)$  representa a porcentagem da diferença (*gap*) entre  $Z_{min}$  e o melhor valor encontrado pelo algoritmo *VNS*, que possui os melhores valores da literatura; a coluna  $Z_{med}$  indica o valor médio da média de 10 execuções de cada instância do grupo, já a coluna  $T$  representa o tempo computacional médio, em segundos, para encontrar a melhor solução nas 10 instâncias do grupo. Os outros 4 grupos de colunas são formados pelas colunas  $Z$  indicando o valor médio das melhores soluções encontradas por cada algoritmo em cada grupo e  $T$  o tempo computacional médio, em segundos, para o algoritmo encontrar sua melhor solução.

Na Tabela 1 podemos observar que o MSLB empatou seu valor ótimo com a melhor solução da literatura (*VNS*) em 11 de 12 grupos de 10 instância distintas. O custo médio da solução de 10 execuções do MSLB empatou em 10 dos 12 grupos com o *VNS*, demonstrando uma boa convergência do algoritmo proposto para os grupos de instâncias com  $|V| = 200$ .

A Tabela 2 apresenta os valores das soluções encontradas pelos algoritmos para instâncias com  $|V| = 500$ . Nela podemos observar que os melhores valores obtidos pelo MSLB empatou em 8 dos 12 grupos de instâncias com o melhor resultado da literatura (*VNS*), e ainda encontrou melhores resultados para 4 grupos de instâncias, chegando a melhorar em 2,1% as soluções encontradas para o grupo 500 – 500 – *hd*. O MSLB em suas soluções médias obteve 4 vitórias, 4 empates e 4 derrotas comparadas as soluções do *VNS*, novamente apresentando uma boa convergência dos resultados médios do MSLB.

O tempo computacional  $T$  apresentado nas Tabelas 1 e 2 é referente ao tempo necessário para que cada algoritmo encontre sua melhor solução. Podemos observar nas

Instância			MSLB				VNS		GRASP		Pilot		MGA	
$ V $	$ L $	$d$	$Z_{min}$	$\Delta(\%)$	$Z_{med}$	$T$	$Z$	$T$	$Z$	$T$	$Z$	$T$	$Z$	$T$
500	625	hd	<b>5,1</b>	0	<b>5,1</b>	2,88	<b>5,1</b>	16,1	<b>5,1</b>	4,9	<b>5,1</b>	184	5,4	51,9
		md	<b>7,8</b>	<b>-1,3</b>	7,84	18,3	7,9	44,7	7,9	35,7	8,1	201	8,3	222,2
		ld	<b>18,1</b>	<b>-1,2</b>	18,19	28,8	18,3	155	18,4	53,1	18,5	290	19,1	297,8
500	500	hd	<b>4,6</b>	<b>-2,1</b>	4,69	3,85	4,7	22,3	4,7	28,1	4,8	200	4,7	117,5
		md	<b>6,5</b>	0	<b>6,5</b>	13,2	<b>6,5</b>	32,3	<b>6,5</b>	90,9	6,7	190	7,1	170,9
		ld	<b>15,8</b>	0	15,86	6,15	<b>15,8</b>	140	15,9	20,2	15,9	300	16,6	241,8
500	250	hd	<b>3</b>	0	3,03	1,99	<b>3</b>	0,14	<b>3</b>	0,25	<b>3</b>	35,5	<b>3</b>	0,52
		md	<b>4,1</b>	0	4,15	8,59	<b>4,1</b>	84	4,2	0,58	4,2	65,3	4,3	28
		ld	<b>9,8</b>	<b>-1</b>	9,8	11,1	9,9	5,1	9,9	3,3	9,9	156	10,1	181,2
500	125	hd	<b>2</b>	0	2,01	1,37	<b>2</b>	0,02	<b>2</b>	0,15	<b>2</b>	3,4	<b>2</b>	0,02
		md	<b>2,6</b>	0	<b>2,6</b>	1,15	<b>2,6</b>	1,1	<b>2,6</b>	0,45	<b>2,6</b>	6,6	<b>2,6</b>	2,6
		ld	<b>6,2</b>	0	<b>6,2</b>	3,25	<b>6,2</b>	3,9	<b>6,2</b>	4	6,3	11,9	<b>6,2</b>	57,1

**Tabela 2. Comparação dos resultados computacionais dos algoritmos aproximativos para instâncias com  $|V| = 500$ .**

Tabelas que o tempo utilizado pelo MSLB para encontrar sua melhor solução é inferior ao utilizado pelo VNS em praticamente todas os grupos de instâncias com  $|L| \geq |V|$  que são consideradas as instâncias mais difíceis. Entretanto, esta comparação não é justa pois a máquina utilizada na literatura tem capacidade de processamento inferior à máquina utilizada na execução do MSLB. Este tempo apresentado serve como indício de que mesmo com a adoção de técnicas como o *Local Branching*, o tempo computacional do MSLB não diverge tanto das outras abordagens aproximativas da literatura. Outra consideração a ser feita é que em todas as instâncias que obtivemos novas melhores soluções, sempre estivemos muito longe do tempo limite, o que demonstra que os algoritmos da literatura não conseguem convergir para as soluções do MSLB mesmo executando por mais tempo.

## 6. Conclusões e Trabalhos Futuros

Neste trabalho abordamos o Problema de Árvore Geradora de Rótulação Mínima (PAGRM), onde dado um grafo não direcionado  $G = (V, E, L)$  pretende-se minimizar o número de rótulos de um conjunto  $L$  associados às arestas  $e \in E$ , ou seja, escolher um subconjunto de arestas que deixe o subgrafo solução conexo e que induza o uso da menor quantidade de rótulos de  $L$ .

Propomos para a resolução do PAGRM uma nova formulação matemática para o PAGRM; um *framework* de planos de corte para a resolução da formulação proposta; uma nova estrutura de vizinhança exata baseada no conceito de *Local Branching*; e uma abordagem híbrida (MSLB) que combina características das meta-heurísticas *Multi Start* e VND com a nova estrutura de vizinhança proposta.

Os experimentos computacionais realizados demonstraram que o MSLB alcançou as melhores soluções da literatura para todos os grupos de instâncias de teste (exceto o grupo 200 – 100 –  $ld$ ), conseguindo avançar as melhores soluções conhecidas para os grupos de instâncias: 500 – 625 –  $ld$ ; 500 – 625 –  $md$ ; 500 – 500 –  $hd$  e 500 – 250 –  $ld$ . Estes resultados se devem à capacidade de diversificação apresentada pela heurística construtiva MS\_MVCA e de intensificação propiciada pela nova estrutura de vizinhança proposta.

Os próximos passos pretendidos para a pesquisa são a exploração das vizinhanças  $LB_4$  e  $LB_5$ , algoritmos de separação para as soluções fracionadas do FBCC durante a resolução do LP, permitindo assim encontrar componentes conexas maiores na soluções inteiras e a inclusão de estruturas de vizinhança heurísticas no VND para aplicar nas instâncias menores afim de diminuir o tempo computacional gasto em suas resoluções.

## Referências

- 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.
- Chimani, M., Kandyba, M., Ljubić, I. e Mutzel, P.** (2010), Obtaining optimal k-cardinality trees fast. *J. Exp. Algorithmics*, v. 14, p. 5:2.5–5:2.23.
- Chwatal, A. M. e Raidl, G. R.** (2011), Solving the minimum label spanning tree problem by mathematical programming techniques. *Adv. Operations Research*, v. 2011.
- 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.
- Hansen, P., Mladenovic, N. e Moreno Perez, J.** (2008), Variable neighbourhood search: methods and applications. *4OR: A Quarterly Journal of Operations Research*, v. 6, p. 319–360. 10.1007/s10288-008-0089-1.
- 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.
- Subramanian, A., Penna, P. H. V., Uchoa, E. e Ochi, L. S.** (2012), A hybrid algorithm for the heterogeneous fleet vehicle routing problem. *European Journal of Operational Research*, v. 221, n. 2, p. 285 – 295.
- Subramanian, A., Uchoa, E. e Ochi, L. S.** (2013), A hybrid algorithm for a class of vehicle routing problems. *Computers Operations Research*, v. 40, n. 10, p. 2519 – 2531.
- Van-Nes, R.** Design of multimodal transport networks: A hierarchical approach. Delft University Press, Delft, 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.