Formulações e Algoritmos Sequenciais e Paralelos para o Problema da Árvore Geradora de Custo Mínimo com Restrição de Grau Mínimo

Leonardo Conegundes Martinez, Alexandre Salles da Cunha

[1]Departamento de Ciência da Computação – Universidade Federal de Minas Gerais

{leocm,acunha}@dcc.ufmg.br

Abstract. *Given an edge weighted undirected graph G and a positive integer d, the Min-degree Constrained Minimum Spanning Tree Problem (MDMST) consists of finding a minimum cost spanning tree T of G, such that each vertex is either a leaf or else has a degree at least d in T. In this work, we introduce Integer Programming formulations and present sequential and parallel optimization algorithms for the problem. With the proposed methods, several new optimality certificates and best lower and upper bounds for MDMST are provided.*

Resumo. *Dados um grafo não direcionado valorado nas arestas G e um inteiro positivo d, o Problema da Árvore Geradora de Custo Mínimo com Restrição de Grau Mínimo (PAGMGM) consiste em encontrar uma árvore geradora de custo mínimo T de G, tal que o grau de cada vértice em T seja igual a 1 ou maior ou igual a d. Neste trabalho, introduzimos formulações de Programação Inteira e algoritmos de otimização sequenciais e paralelos para o problema. Com os métodos propostos, vários novos certificados de otimalidade e melhores limites inferiores e superiores para o PAGMGM são fornecidos.*

1. Introdução

Árvores geradoras (AGs) são estruturas de fundamental importância em Teoria dos Grafos e no projeto de redes cujas entidades devem se manter conexas. Um dos problemas mais importantes em Otimização Combinatória é o Problema da Árvore Geradora Mínima (PAGM), que consiste em encontrar uma árvore geradora de custo mínimo de um grafo não direcionado valorado nas arestas. Embora o PAGM possa ser resolvido em tempo polinomial, em muitas aplicações é necessário impor restrições adicionais para que topologias como AGs possam ser empregadas no projeto de redes, o que torna em geral o problema resultante difícil de ser resolvido.

O problema estudado neste trabalho, conhecido como *Problema da Árvore Geradora de Custo Mínimo com Restrição de Grau Mínimo* (PAGMGM), pertence a esse grupo de problemas definidos em AGs com restrições adicionais. Seja $G = (V, E)$ um grafo não direcionado, com conjunto de vértices V ($n = |V|$), conjunto de arestas E ($m = |E|$), e custos $\{c_{ij} \in \mathbb{R}_+ : \{i, j\} \in E\}$ associados às arestas. Dado um inteiro positivo $d < n$, o PAGMGM consiste em encontrar uma árvore geradora de custo mínimo T de G, tal que o grau de cada vértice em T seja igual a 1 ou maior ou igual a d.

Dentre os demais problemas definidos em AGs com restrições adicionais, destaca-se a forte relação do PAGMGM com o clássico Problema da Árvore Geradora de Custo Mínimo com Restrição de Grau (PAGMG). Enquanto o PAGMG impõe um limite *superior* no grau dos vértices de uma AG viável, o PAGMGM modela situações em que recursos, informações ou processamento devem ser centralizados em alguns nós especiais (nós centrais), ao invés de serem dispersos pela rede. Para tanto, é imposto um limite *inferior* no grau de cada nó central, de modo a compensar os recursos neles instalados.

Embora tenha sido proposto recentemente, o PAGMGM já despertou considerável interesse da comunidade acadêmica. O problema foi introduzido por [Almeida et al. 2006], em um estudo que demostrou que o PAGMGM é NP-Difícil para $d \geq 4$ (posteriormente foi provado que o problema também é NP-Difícil para $d = 3$ [Almeida et al. 2010]). Os autores apresentaram formulações de Programação Inteira (PI) baseadas em restrições de balanço de fluxo de uma ou múltiplas mercadorias e um algoritmo Branch-and-bound (BB) para a resolução do problema. Novas formulações definidas a partir das desigualdades de Miller-Tucker-Zemlin (MTZ) para eliminação de subcircuitos foram apresentadas por [Akgún and Tansel 2010], que testaram um algoritmo BB baseado na melhor delas. A aplicação de métodos heuríticos para o PAGMGM foi investigada pela primeira vez em [Martins and Souza 2009], em um estudo que apresentou diversos métodos e heurísticas baseados em algoritmos do tipo *second order* e em diferentes implementações da meta-heurística *Variable Neighborhood Search* (VNS).

Neste trabalho, avançamos no estudo sobre o PAGMGM sob uma perspectiva teórica e computacional [Martinez and Cunha 2010, Martinez and Cunha 2011, Martinez and Cunha 2012, Martinez 2012]. As principais contribuições são: (i) introdução de três formulações de PI para o PAGMGM, sendo uma delas a que fornece os limites de Programação Linear mais fortes na literatura do problema; (ii) estudo e aplicação da técnica de Reformulação por Interseção, que pode ser utilizada para reformular outras variações do PAGM, (iii) apresentação de duas novas classes de desigualdades lineares válidas para o PAGMGM; (iv) desenvolvimento de algoritmos de otimização sequenciais e paralelos, a saber: um algoritmo Branch-and-cut, um método Local Branching, uma Relaxação Lagrangeana paralela, uma heurística construtiva multipartida probabilística e uma Heurística Lagrangeana; e (v) obtenção de novos certificados de otimalidade e novos melhores limites inferiores e superiores para várias instâncias do PAGMGM.

2. Formulações de Programação Inteira

Em problemas de otimização em grafos cujas soluções são representadas por AGs sujeitas a restrições complicantes adicionais, é usual definir a formulação a partir de dois conjuntos independentes de restrições, aqui chamados de *restrições de topologia de árvore geradora* e *restrições complicantes adicionais*. Para o caso particular do PAGMGM, o segundo conjunto deve impor as *restrições de grau mínimo* dos vértices centrais.

Para definir o conjunto de restrições de topologia de árvore geradora várias abordagens podem ser utilizadas. Para o PAGMGM, foram apresentadas na literatura formulações baseadas em restrições de balanço de fluxo de uma ou múltiplas mercadorias e em desigualdades MTZ. Uma vantagem dessas formulações é o fato de serem compactas. No entanto, o tamanho das formulações com múltiplas mercadorias as torna inapropriadas para a aplicação direta em procedimentos de enumeração como o BB. Por outro lado, formulações baseadas em fluxos de uma única mercadoria ou nas desigualdades MTZ apresentam em geral limites de Programação Linear (PL) fracos. Introduzimos a seguir três novas formulações de PI para o PAGMGM, duas baseadas em um número exponencial de restrições de eliminação de subcircuitos, uma não direcionada e outra direcionada, e uma terceira formulação baseada em uma técnica denominada Reformulação por Interseção.

2.1. Formulações com um número exponencial de restrições

Uma formulação canônica para o PAGMGM, denotada por P_N, pode ser obtida empregando-se variáveis binárias para selecionar as arestas de E e as folhas de V. Para garantir a topologia de AGs em soluções viáveis empregamos as restrições de eliminação de subcircuitos não direcionadas, denominadas *Subtour Elimination Constrainsts* (SECs).

Outra maneira de formular o PAGMGM consiste em considerar o problema em um grafo direcionado $D = (V, A)$, modelar o conjunto de soluções viáveis como arborescências em D que satisfazem as restrições de grau mínimo e escolher um vértice especial $r \in V$ para representar a raiz da arborescência procurada. Denotamos por P_D^r a formulação direcionada cujas soluções viáveis são enraizadas no vértice $r \in V$. Assim, P_D^r pode ser obtida empregando-se variáveis binárias para selecionar os arcos de A e as folhas de V, e um número exponencial de restrições de eliminação de subcircuitos direcionadas, denominadas *Directed Cutset Constraints* (DCUTs).

Além das restrições de topologia de árvore geradora e de grau mínimo, as formulações P_N e P_D^r foram fortalecidas com outras desigualdades válidas para o politopo do PAGMGM, que estabelecem um limite superior para o número de vértices centrais e impõem condições lógicas associadas à seleção de arestas ou arcos cujas extremidades são vértices folha. Como a formulação P_D^r não é simétrica em relação à escolha da raiz, definimos uma reformulação para o PAGMGM com limites simétricos, discutida a seguir.

2.2. Reformulação por Interseção

Gouveia and Telhada (2008) propuseram uma técnica de reformulação, denominada Reformulação por Interseção (RI), que pode ser empregada para o PAGMGM, fornecendo uma formulação simétrica e com limites de PL mais fortes para o problema. A principal ideia da técnica consiste em considerar simultaneamente todas as formulações P_D^r, para todo $r \in V$, de forma a encontrar n arborescências geradoras de D, cada uma enraizada em um vértice diferente de V, tal que, desconsiderando a orientação dos arcos, todas utilizem as mesmas arestas de G. Isso significa dizer que se o arco (i, j) é incluído em uma solução do PAGMGM formulado com $P_D^{r_1}$, um dos arcos (i, j) ou (j, i) deve ser incluído na solução de outra formulação $P_D^{r_2}$, para todo $r_2 \neq r_1$. Na reformulação P_I, empregamos variáveis binárias $\{z_{ij} \in \mathbb{B} : \{i, j\} \in E\}$ para selecionar arestas, $\{y_i \in \mathbb{B} : i \in V\}$ para selecionar vértices e $\{x_{ij}^r \in \mathbb{B} : (i, j) \in A, r \in V\}$ para selecionar arcos nas arborescências enraizadas em $r \in V$.

Considere as seguintes notações e definições. Dada uma formulação P para o PAGMGM, assuma que $w(P)$ indique o limite de PL fornecido por P. Dado um grafo não direcionado $G = (V, E)$ e um conjunto não vazio $W \subseteq V$, $\delta(W) := \{\{i, j\} \in E : i \in W, j \notin W\}$. Dado um grafo direcionado $D = (V, A)$ e um conjunto não vazio $W \subset V$, $\delta^-(W) := \{(i, j) \in A : i \notin W, j \in W\}$ e $A(W) := \{(i, j) \in A : i \in W, j \in W\}$. Para qualquer função real $f : Q \to \mathbb{R}$ definida em um domínio finito Q e $\overline{Q} \subseteq Q$, considere que $f(\overline{Q}) := \sum_{q \in \overline{Q}} f_q$. Assim, a reformulação por interseção para o PAGMGM é dada por:

$$w = \min \left\{ \sum_{\{i,j\} \in E} c_{ij} z_{ij} : (z, x, y) \in P_I \cap (\mathbb{B}^m, \mathbb{B}^{2mn}, \mathbb{B}^n) \right\}, \quad (1)$$

onde o poliedro $P_I \subset \mathbb{R}^{m(2n+1)+n}$ é definido por:

$$x^r(A) = n-1, \forall r \in V, \quad (2)$$

$$x^r(\delta^-(i)) = 1, \forall r \in V, i \in V \setminus \{r\}, \quad (3)$$

$$z(\delta(i)) \geq 1 + (d-1)(1-y_i), \forall i \in V, \quad (4)$$

$$z(\delta(i)) \leq 1 + (n-2)(1-y_i), \forall i \in V, \quad (5)$$

$$z_{ij} + y_i + y_j \leq 2, \forall \{i,j\} \in E, \quad (6)$$

$$x^r_{ij} + y_i \leq 1, \forall r \in V, (i,j) \in A \backslash \delta^+(r), \quad (7)$$

$$|V| - y(V) \leq \lfloor (n-2)/(d-1) \rfloor, \quad (8)$$

$$z_{ij} = x^r_{ij} + x^r_{ji}, \forall r \in V, \forall \{i,j\} \in E, \quad (9)$$

$$z_{ij} \geq 0, \forall \{i,j\} \in E, \quad (10)$$

$$0 \leq y_i \leq 1, \forall i \in V, \quad (11)$$

$$0 \leq x^r_{ij} \leq 1, \forall r \in V, (i,j) \in A. \quad (12)$$

As restrições (2) e (3) são parte do conjunto de restrições de topologia de arborescência geradora; (4) e (5) definem o conjunto de restrições de grau mínimo; (6) e (7) impõem condições lógicas associadas à seleção de arestas ou arcos cujas extremidades são vértices folha; (8) estabelece um limite superior para o número de vértices centrais e (9) estabelecem a interseção entre as formulações. Resultados teóricos apresentados em [Martinez 2012] mostraram que as SECs ou DCUTS são redundantes para P_I e que $w(P_I) \geq w(P^r_D) \geq w(P_N), \ \forall r \in V$.

2.3. Experimentos computacionais

Em todos os experimentos foram utilizadas 90 instâncias definidas em grafos completos de 30 a 500 vértices e $d \in \{3, 5, 10, 20\}$, que correspondem a todas as instâncias do PAGMGM já testadas na literatura. Para avaliar computacionalmente os limites de PL das formulações P_N, P^r_D e P_I, implementamos algoritmos de planos de corte, enquanto $w(MTZ)$ foi avaliado diretamente pelo *solver* de PL do CPLEX 10.2. Os limites fornecidos pela formulação mais forte proposta em [Almeida et al. 2006] são iguais aos limites fornecidos por P^r_D (considerando uma mesma raiz), e por isso não são discutidos a seguir.

Os tempos computacionais necessários para avaliar os limites de PL das formulação MTZ, P_N e P^r_D foram baixos e apresentaram a mesma ordem de magnitude. Já os tempos computacionais para avaliar $w(P_I)$ foram elevados, sendo até quatro ordens de magnitude maiores para algumas instâncias. No entanto, os limites fornecidos pela formulação P_I são muito mais fortes que os fornecidos pelas demais formulações. Considerando apenas as 42 instâncias para as quais foi possível avaliar $w(P_I)$ no tempo imposto, os *gaps* de dualidade médios das formulações MTZ, P_N, P^r_D e P_I foram iguais a 8,12%, 8,82%, 5,92% e 2,81%, respectivamente. Além disso, em cinco casos $w(P_I)$ foi igual ao custo da solução ótima, e em todos eles as soluções de PL foram inteiras.

3. Algoritmos de Otimização

Nesta seção introduzimos métodos exatos e heurísticos para a resolução do PAGMGM. Apesar da formulação P_I fornecer melhores limites inferiores que a formulação P^r_D, na prática, os tempos computacionais gastos para availá-los tornam proibitivo o uso da RI em um procedimento enumerativo inteligente, do tipo BB, se nenhum tratamento adicional for feito. Em uma primeira tentativa de resolver o PAGMGM, introduzimos dois algoritmos de resolução exata baseados na formulação P^r_D, a saber: um algoritmo Branch-and-cut e um algoritmo Local Branching, resumidos a seguir.

Algoritmo Branch-and-cut: Na nossa implementação do método Branch-and-cut (BC), utilizamos o *solver* de PI do CPLEX para fazer chamadas ao algoritmo de planos de corte que avalia o limite de PL da formulação P^r_D. As DCUTS foram separadas pela resolução de problemas de fluxo máximo e as restrições lógicas por inspeção. A subdivisão do espaço de busca na árvore de enumeração foi realizada com maior prioridade associada às ramificações nas variáveis y, de acordo com a opção *strong branching* do CPLEX.

Algoritmo Local Branching: O Local Branching (LB) é um algoritmo exato que pode ser interpretado como uma formalização de procedimentos de busca local através de modelos de Programação Matemática. A nossa implementação do algoritmo utiliza a formulação P_D^r, empregando o algoritmo BC para resolver os subproblemas gerados pelo método a partir da restrição de LB definida nas variáveis y.

Relaxação Lagrangena: Dadas a dificuldade em se resolver as maiores instâncias do PAGMGM pelos métodos BC e LB e a qualidade dos limites de PL fornecidos por P_I, propusemos uma Relaxação Lagrangeana (RL) baseada nessa formulação, com o objetivo de avaliar mais rapidamente $w(P_I)$. Para tanto, relaxamos e dualizamos as restrições (4)-(7) e (9), obtendo um Problema de Relaxação Lagrangeana que pode ser decomposto em $n+2$ subproblemas independentes muito simples e de resolução eficiente: um envolvendo a variável z, outro envolvendo a variável y e um envolvendo cada variável $\{x^r : r \in V\}$. O Problema Dual Lagrangeano foi resolvido com uma implementação paralela em `OpenMP` do *Deflected Subgradient Method* (DSM), avaliado em um processador com 6 núcleos.

Heurística Lagrangena: Pelo lado primal, desenvolvemos uma Heurística Lagrangeana paralela com o objetivo de obter melhores limites superiores para o PAGMGM. A heurística envolve duas fases, uma construtiva e outra de melhoria. Na primeira fase, implementamos uma variação de uma heurística construtiva probabilística multipartida que propusemos para o PAGMGM. Na fase de melhoria, empregamos o algoritmo LB descrito anteriormente, formulando a restrição de Local Branching a partir da melhor solução viável para o PAGMGM encontrada na fase construtiva.

3.1. Experimentos Computacionais

Os métodos BC e LB dominaram os métodos BB-MTZ [Akgún and Tansel 2010] e HVNS [Martins and Souza 2009] em termos de tempos computacionais e qualidade dos limites superiores encontrados. Em média, os métodos obtiveram um tempo computacional entre uma e duas ordens de grandeza inferior ao tempo gasto pelo BB-MTZ para resolver as instâncias de 30 a 100 vértices. Para as instâncias maiores, as soluções fornecidas pelos métodos BC e LB foram em média quase 10% melhores que as obtidas pelas meta-heurísticas HVNS. Considerando as instâncias em que o método de planos de corte não conseguiu avaliar $w(P_I)$ dentro do limite de tempo imposto, os limites inferiores dados pela RL foram, em média, $2{,}90\%$ mais fortes, além de terem sido obtidos em $0{,}96\%$ do tempo computacional. Em 14 das 36 instâncias em aberto do PAGMGM, a RL forneceu melhores limites inferiores que aqueles fornecidos ao final dos algoritmos BC e LB, obtidos após a exploração de milhares de nós em suas árvores de enumeração. Sob uma perspectiva de programação paralela, bons níveis de eficiência foram alcançados, principalmente para as instâncias de maior porte (*speedup* superior a 4).

A Tabela 1 mostra uma comparação entre o número de melhores limites inferiores e superiores conhecidos para o PAGMGM obtidos pelos métodos HVNS, BB-MTZ, BC, LB e RL, para cada tamanho de grafo associado às 36 instâncias do problema para as quais nenhum método avaliado foi capaz de prover o certificado de otimalidade.

Tabela 1. # de melhores limites inferiores e superiores fornecidos pelos métodos.

n	#instâncias em aberto	Melhores limites inferiores				Melhores limites superiores				
		BB-MTZ	BC	LB	RL	HVNS	BB-MTZ	BC	LB	RL
100	6	0	6	0	0	0	0	0	6	4
200	6	0	0	0	6	0	0	0	1	5
300	6	0	0	0	6	0	0	0	1	5
400	9	1	6	0	2	0	2	1	1	5
500	9	1	8	0	0	0	2	3	3	1

4. Conclusões e Trabalhos Futuros

Neste trabalho, sumarizamos as principais contribuições da nossa pesquisa em formulações e algoritmos de otimização exatos e heurísticos para o PAGMGM. Resultados teóricos e computacionais mostraram que as formulações e os métodos propostos conseguiram melhorar resultados da literatura, fornecendo os melhores limites de PL para o problema, assim como 6 novos certificados de otimalidade, 34 melhores limites inferiores e 32 melhores limites superiores. No entanto, ainda encontramos dificuldades para a resolução das maiores instâncias do problema. Como trabalhos futuros, pretendemos realizar um estudo poliedral, implementar versões paralelas do LB e da RL para sistemas de memória distribuída e compartilhada e desenvolver um algoritmo BB para a resolução exata do problema, baseado na RL. Devido ao restrito espaço, omitimos neste texto importantes referências e resultados da pesquisa (por favor veja [Martinez 2012]).

Referências

Akgún, I. and Tansel, B. (2010). Min-degree constrained minimum spanning tree problem: New formulation via Miller-Tucker-Zemlin constraints. *Computers & Operations Research*, 37(1):72–82.

Almeida, A. M., Martins, P., and Souza, M. C. (2006). Min-Degree Constrained Minimum Spanning Tree Problem: Complexity, proprieties and formulations. Technical Report 6/2006, Centro de Investigação Operacional, Universidade de Lisboa.

Almeida, A. M., Martins, P., and Souza, M. C. (2010). md-MST is NP-hard for $d \geq 3$. *Electronic Notes in Discrete Mathematics*, 36:9–15.

Gouveia, L. and Telhada, J. (2008). The multi-weighted Steiner Tree problem: A reformulation by intersection. *Computers & Operations Research*, 35:3599–3611.

Martinez, L. C. (2012). Formulações e algoritmos sequencias e paralelos para o problema da Árvore geradora de custo mínimo com restrição de grau mínimo. Master's thesis, Departamento de Ciência da Computação, Universidade Federal de Minas Gerais.

Martinez, L. C. and Cunha, A. S. (2010). Finding min-degree constrained spanning trees faster with a branch-and-cut algorithm. *Electronic Notes in Discrete Mathematics*, 36:311–318. ISCO - International Symposium on Combinatorial Optimization.

Martinez, L. C. and Cunha, A. S. (2011). The Min-Degree Constrained Minimum Spanning Tree Problem: Formulations and Branch-and-cut algorithm. *Discrete Applied Mathematics*. In Press, doi:10.1016/j.dam.2011.08.008.

Martinez, L. C. and Cunha, A. S. (2012). A parallel lagrangian relaxation algorithm for the min-degree constrained minimum spanning tree problem. *Lecture Notes in Computer Science*, 7422:237–248.

Martins, P. and Souza, M. C. (2009). VNS and second order heuristics for the min-degree constrained minimum sp anning tree problem. *Computers & Operations Research*, 36:2669–2982.