

Uma Variante do Problema da Árvore Geradora Mínima Generalizado com Prêmios nos Vértices

Emanuella Toledo Lopes, Luiz Satoru Ochi, Yuri Abitbol de Menezes Frota

Instituto de Computação, Universidade Federal Fluminense

Rua Passo da Pátria, 156 - Bloco E - 3º andar, São Domingos, Niterói - RJ

{etoledo,satoru,yuri}@ic.uff.br

RESUMO

O Problema da Árvore Geradora Mínima Generalizado (PAGMG) pode ser modelado sobre um grafo G não direcionado e ponderado cujos vértices estão divididos em componentes. O objetivo do PAGMG é encontrar uma árvore T que cubra exatamente um vértice de cada componente de G , de forma que a soma dos custos das arestas de T seja mínima. A versão *at Least* do PAGMG, L-PAGMG, objetiva encontrar uma árvore T que cubra pelo menos um vértice de cada componente de G . O PAGMG com Coleta de Prêmios (CP-PAGMG), baseado no PAGMG, objetiva minimizar a diferença do custo total das arestas utilizadas pela soma total dos prêmios coletados nos vértices selecionados. Neste trabalho propomos uma nova variante para o PAGMG com prêmios nos vértices (PAGMG-P). Sua definição é baseada no L-PAGMG, com a característica adicional de que existe um prêmio associado a cada vértice e cada componente possui um prêmio mínimo a ser coletado. Não foram encontrados trabalhos sobre o PAGMG-P na literatura. Dessa forma, propomos uma formulação matemática e geramos um conjunto de instâncias para o problema. São apresentados testes computacionais que ilustram a eficiência do método exato proposto.

PALAVRAS CHAVE Problema da Árvore Geradora Mínima Generalizado. Formulação Matemática. Programação Matemática.

Área principal PM - Programação Matemática. OC - Otimização Combinatória.

ABSTRACT

The Generalized Minimum Spanning Tree Problem (GMSTP) can be modeled on a weighted and undirected graph G whose vertices are divided in components. GMSTP aims to find a tree T that covers exactly one vertex of each component of G , in a way that sum edges cost of T be minimum. The GMSTP *at Least* version, L-GMSTP, aims to find a tree T that covers at least one vertex of each component of G . The Prize-Collecting GMSTP (PC-GMSTP), based on GMSTP, aims to minimize the difference of edges total cost used by the sum total of collected prizes in the selected vertices. In this paper we propose a new GMSTP variant with prizes in the vertices (P-GMSTP). It is based on L-GMSTP, with the additional feature that there is a prize associated to each vertex and each component has a minimum prize to be collected. There were not found studies about P-GMSTP in the literature. Thus, we propose a mathematical formulation and we generate a instance set to this problem. Computational experiments are presented to illustrate the efficiency of the exact method proposed.

KEYWORDS Generalized Minimum Spanning Tree Problem. Mathematical Formulation. Mathematical Programming.

Main area PM - Mathematical Programming. OC - Combinatorial Optimization.

1 Introdução

Este trabalho aborda uma variante do Problema da Árvore Geradora Mínima Generalizado (PAGMG), em que o conjunto de vértices está dividido em componentes disjuntas e o objetivo é determinar uma árvore de custo mínimo que cubra exatamente um vértice de cada componente. O problema foi introduzido e provado ser \mathcal{NP} -Difícil por Myung et al. (1995).

O PAGMG pode ser definido da seguinte forma. Seja um grafo não direcionado e ponderado $G = (V, E)$, onde V representa o conjunto dos vértices e E o conjunto das arestas. Cada aresta $e \in E$ possui um custo associado $c_e \in \mathcal{R}^+$. O conjunto de vértices V é particionado em m componentes disjuntas e não vazias V_1, \dots, V_m . O objetivo é determinar uma árvore de custo mínimo que cubra exatamente um vértice de cada componente (Ferreira, 2007).

Uma outra versão do PAGMG abordada na literatura é o problema de encontrar uma árvore que cubra pelo menos um vértice de cada componente, denominada L-PAGMG. Esse problema foi introduzido por Ihler et al. (1999) com outra denominação, Problema de Steiner em Grupos. Foi provado que o problema é um caso particular do Problema de Steiner em Grafos Generalizado (Salazar, 1991).

Golden et al. (2007) propuseram uma variante para o PAGMG, o Problema da Árvore Geradora Mínima Generalizado com Coleta de Prêmios (CP-PAGMG), que é definido de maneira semelhante ao problema original, mas tem a característica adicional de que os vértices em cada componente competem entre si, oferecendo um prêmio se o vértice for selecionado. O objetivo é minimizar a diferença do custo total das arestas utilizadas pela soma total dos prêmios coletados nos vértices selecionados.

Neste trabalho, propomos uma outra variante do PAGMG que, assim como o CP-PAGMG, possui prêmios nos vértices, mas que se difere do modelo proposto em Golden et al. (2007) tanto em restrições quanto em objetivo. A variante aqui proposta exige que seja adquirido um prêmio mínimo P_i em cada componente V_i e, enquanto o CP-PAGMG restringe a escolha de exatamente um vértice por componente, na variante proposta pode ser escolhido pelo menos um vértice. Além disso, o objetivo da nova versão é minimizar o custo total da árvore solução, sem considerar os prêmios coletados.

A justificativa de se propor uma nova variante do PAGMG com prêmios nos vértices, é a de que em diversas possíveis aplicações reais é exigido um prêmio (ou capacidade) mínimo para cada componente V_i (Santos et al., 2012).

No contexto de aplicações para o PAGMG, estas podem ser encontradas na área de projeto de síntese de redes de telecomunicações, por exemplo, onde as redes regionais precisam ser interconectadas por uma árvore que contenha uma conexão para cada sub-rede. Para essa interconexão, um vértice de cada sub-rede deve ser selecionado como *gateway* (Myung et al., 1995).

Outra aplicação do PAGMG está relacionada com vários problemas de localização de instalações que precisam estar conectadas por meio de rodovias ou *links* de comunicação (Feramans et al., 2000; Kansal & Torquato, 2001). Um exemplo acontece quando uma empresa quer estabelecer centros regionais de distribuição para suas lojas e precisa selecionar um ponto em cada região a fim de construir uma rede de comunicação interconectando esses centros (Shyu et al., 2006).

Uma aplicação do L-PAGMG, descrita por Dror et al. (2000), pode ser encontrado no projeto de redes de irrigação em ambientes desérticos, onde a escolha de pontos de irrigação deve ser feita de maneira que nenhuma parte do ambiente fique sem receber irrigação. Assim, a área a ser irrigada é dividida em áreas menores e o objetivo é encontrar uma rede de irrigação mínima que cubra todas as áreas, cada área com pelo menos um ponto de irrigação.

O CP-PAGMG pode ser aplicado no mesmo contexto do primeiro exemplo descrito para o PAGMG. No entanto, neste caso, os vértices a serem selecionados oferecem uma compensação monetária, ou um “prêmio”, se selecionado como o nó *gateway* para a sua região. O objetivo é minimizar o custo total de *links* usados para conectar as regiões compensando com a soma total de prêmios coletados a partir dos nós *gateway* selecionados para o projeto (Golden et al., 2007).

Exemplos de compensação monetária na seleção de nós em uma rede de telecomunicações surge em vários contextos do mundo real. Por exemplo, no projeto de redes de cabos submarinos interligando continentes diferentes: nem todos os países ou cidades inclusos na rota da rede podem ser conectados diretamente a ela. Isso se deve aos altos custos de um local ser ligado a uma rede deste porte. Conseqüentemente, os planejadores destas redes geralmente designam um local a ser selecionado a partir de um determinado conjunto de regiões que fazem parte de sua rota. Dados os potenciais benefícios monetários associados a ser um local que está diretamente ligado a uma rede transcontinental de fibra óptica, com significativos benefícios econômicos, não é incomum que as cidades ou países disputem uns contra os outros a seleção como um local de interconexão. Estes incentivos monetários são geralmente sob a forma de créditos fiscais e abatimentos para o construtor ou operador da rede de telecomunicações (Golden et al., 2007).

1.1 Motivação

Considerando as versões do PAGMG apresentadas neste trabalho, pode-se afirmar que estas são próximas a uma grande quantidade de problemas na vida real. No entanto, elas ainda não se aplicam a algumas classes de problemas de redes mais específicos como, por exemplo, o problema de projetar uma rede de internet cabeada para uma cidade com o intuito de oferecer internet *wireless* à população. Sabendo que geralmente uma cidade é dividida em regiões, suponha que a empresa contratada para fazer este serviço deve atender pelo menos uma certa porcentagem da população de cada região de forma obrigatória e que a empresa utilizará pontos de acesso homogêneos em cada região de acordo com a demanda. As regiões possuem tamanho e número de moradores distintos, logo o número mínimo de clientes a serem atendidos difere de uma região para outra. Dessa forma, o objetivo é encontrar uma rota de custo mínimo para a rede cabeada que contenha um ou mais pontos de acesso à rede *wireless* para os moradores de cada região, e a soma dos moradores atendidos pelos pontos de acesso da região deve ser no mínimo uma dada porcentagem de moradores da mesma região.

Modelando o exemplo do problema dado para o PAGMG, temos que as regiões da cidade são as componentes, os pontos de acesso à rede *wireless* são os vértices e a árvore a ser minimizada é formada pela rede cabeada, onde o comprimento de cada cabo é o custo de cada aresta. Seguindo a definição do PAGMG, o objetivo é encontrar uma árvore de custo mínimo que cubra exatamente um vértice de cada componente. Porém, o objetivo do exemplo dado não está sendo atendido, pois o objetivo do PAGMG busca apenas minimizar o custo da solução, além de somente poder escolher um único vértice para cada componente. Apesar de no L-PAGMG poder escolher mais de um vértice por componente, nele a escolha do vértice é feita considerando apenas a minimização dos custos, assim como no PAGMG. Logo, essa versão não atende ao exemplo. No mesmo contexto, pode-se observar que o CP-PAGMG também não atende ao objetivo do exemplo dado, visto que nele pode-se escolher apenas um vértice por componente, e como as regiões possuem demandas diferentes, certamente alguma das regiões deixaria de ser atendida.

Dessa forma, observa-se a necessidade de definir problemas que sejam mais específicos e que possam representar ainda mais problemas da vida real. Com este intuito, propomos uma nova variante para o Problema da Árvore Geradora Mínima Generalizado com Prêmios nos Vértices.

Este trabalho está organizado da seguinte forma. A próxima seção traz a definição da vari-

ante proposta. Na Seção 3 é descrita a formulação matemática proposta. Na Seção 4, primeiramente são apresentadas as ferramentas utilizadas; em seguida, apresenta-se a forma das instâncias criadas; e, por fim, os resultados experimentais alcançados são exibidos. A Seção 5 aborda as conclusões finais e aponta as direções para os trabalhos futuros que serão realizados na continuação deste trabalho.

2 PAGMG com prêmios nos vértices

O PAGMG com prêmios nos vértices (PAGMG-P) aqui proposto é uma variante do PAGMG e sua definição é baseada no L-PAGMG, com a característica adicional de que cada vértice $v \in V$ possui um prêmio associado $p_v \in \mathcal{R}^+$ e cada componente V_i possui um prêmio mínimo a ser adquirido.

Formalmente, o PAGMG-P é definido como segue. Seja um grafo não direcionado e ponderado $G = (V, E)$, onde V representa o conjunto dos vértices e E o conjunto das arestas. Para todo vértice $v \in V$, definimos $N(v) = \{J \mid (v, J) \text{ e } (J, v) \in E\}$ como o conjunto de vértices vizinhos a v . Cada aresta $e \in E$ possui um custo associado $c_e \in \mathcal{R}^+$ e cada vértice $v \in V$ possui um prêmio associado $p_v \in \mathcal{R}^+$. O conjunto de vértices V é particionado em m componentes disjuntas e não vazias V_1, \dots, V_m . O objetivo do PAGMG-P é determinar uma árvore de custo mínimo que cubra pelo menos um vértice de cada componente V_i , cuja soma dos prêmios dos vértices de cada componente V_i pertencentes a solução deve ser maior ou igual a um valor predefinido P_i .

Diferentemente do PAGMG, que é amplamente abordado na literatura, de nosso conhecimento não existem trabalhos sobre o PAGMG-P aqui proposto até o presente momento.

Modelando o exemplo dado na Seção 1.1, temos as componentes representando as regiões da cidade, os vértices representando os pontos de acesso à rede *wireless* e as arestas representando os cabos de rede e formando a árvore a ser minimizada. O número de moradores atendidos por um ponto de acesso é representado pelo valor do prêmio no vértice, e o número mínimo obrigatório de moradores a ser atendido em uma região é representado pelo prêmio mínimo da componente. Assim, o objetivo é determinar uma rede cabeada de custo mínimo que tenha pelo menos um ponto de acesso à rede *wireless* em cada região, sendo que a soma dos moradores atendidos pelos pontos de acesso em uma dada região deve ser maior ou igual ao número mínimo obrigatório de moradores atendidos da mesma região.

Uma aplicação interessante da literatura para o modelo aqui proposto é encontrado no Instituto Brasileiro de Geografia e Estatística (IBGE), onde um grafo (região) é particionado em componentes (cidades, bairros), cada uma com uma capacidade mínima exigida (Semaan et al., 2010).

Um exemplo de instância para a variante proposta do PAGMG-P é ilustrado na Figura 1. Cada vértice v é associado a um prêmio $p_v \geq 0$. Na figura, esta relação é representada por v/p_v . Assim, um dado vértice v possui o prêmio p_v . Em cada componente V_i , a soma dos prêmios nos vértices presentes na árvore solução deve ser no mínimo P_i e o custo total da árvore deve ser minimizado. No exemplo, uma solução obtida pode ser representada pelas arestas $(5, 3)$, $(3, 2)$, $(2, 7)$, $(7, 8)$, $(7, 10)$, $(7, 11)$ e $(11, 12)$.

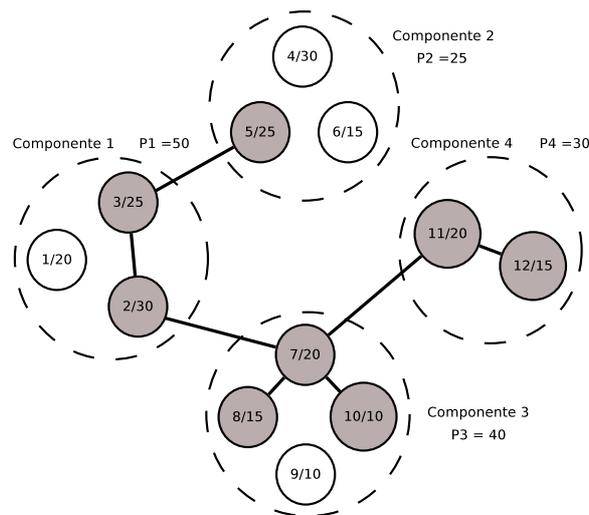


Figura 1: Um exemplo de solução para o PAGMG-P.

3 Formulação matemática para o PAGMG-P

Ainda não se tem conhecimento de trabalhos sobre o PAGMG-P aqui abordado, logo não é conhecida uma formulação matemática para o problema na literatura. Dessa forma, propomos uma formulação para o PAGMG-P baseada na formulação para o CP-PAGMG, apresentada por Golden et al. (2007). Tal formulação é descrita a seguir.

$$\text{Minimizar } \sum_{\forall (i,j) \in E} c_{ij} u_{ij} \quad (1)$$

Sujeito a:

$$\sum_{i \in V_k} z_i \geq 1 \quad \forall k = 1, \dots, m, \quad (2)$$

$$\sum_{(i,j) \in E} u_{ij} = \sum_{i \in V} z_i - 1, \quad (3)$$

$$\sum_{i \in V_k} p_i z_i \geq P_k \quad \forall k = 1, \dots, m, \quad (4)$$

$$u_{ij} \leq z_i \quad \forall (i, j) \in E, i \in V, \quad (5)$$

$$u_{ij} \leq z_j \quad \forall (i, j) \in E, j \in V, \quad (6)$$

$$u_{ij} = w_{kij} + w_{kji} \quad \forall (i, j) \in E, k \in V, \quad (7)$$

$$\sum_{j \in N(i)} w_{kij} \leq 1 \quad \forall k \in V, i \in V - \{k\}, \quad (8)$$

$$w_{kkj} = 0 \quad \forall k \in V, j \in N(k), \quad (9)$$

$$u_{ij} \in \{0, 1\} \quad \forall (i, j) \in E, \quad (10)$$

$$z_i \in \{0, 1\} \quad \forall i \in V, \quad (11)$$

$$w_{kij} \in \{0, 1\} \quad \forall i, j, k \in V. \quad (12)$$

Na formulação representada por (1 – 12): c_{ij} representa o custo fixo da aresta (i, j) ; u_{ij} são variáveis binárias e recebem valor 1 se a solução contém a aresta entre os vértices i e j , e valor 0 caso contrário; se o vértice i pertence a solução, então a variável binária z_i vale 1, caso contrário z_i vale 0; w_{kij} são variáveis binárias que valem 1 se o vértice j é um antecessor do vértice i na árvore direcionada com raiz no vértice k , e valem 0 caso contrário; p_i representa o prêmio fixo associado ao vértice z_i e P_k representa o prêmio mínimo fixo a ser adquirido na componente V_k .

Em (1) temos a função objetivo que requer a minimização da árvore solução. As restrições (2) e (3) garantem que a árvore solução cobre pelo menos um vértice de cada componente. O conjunto de restrições (4) garante que o somatório dos prêmios dos vértices selecionados em cada componente V_k seja pelo menos P_k . As restrições (5) e (6) garantem que se uma aresta pertence à solução, então os vértices que a compõem também pertencem. O conjunto de restrições (7) garante que na árvore direcionada com raiz no vértice k , ou o vértice j é antecessor do vértice i , ou o vértice i é antecessor do vértice j (i. e. ou o vértice j é “pai” de i na árvore ou vice-versa). As restrições (8) garantem que cada vértice tem no máximo um “pai”, exceto o vértice raiz. O conjunto de restrições (9) garante que o vértice raiz não possui “pai”, ou seja, ele não possui antecessor. As restrições (7), (8) e (9) são as restrições de eliminação de sub-ciclos. E, por fim, as restrições (10), (11) e (12) descrevem a integralidade das variáveis u , z e w .

Sobre as restrições de eliminação de sub-ciclos, temos que cada conjunto de variáveis w_{kij} para um determinado vértice k representa um conjunto de variáveis que define uma árvore direcionada com raiz no vértice k , onde uma variável individual w_{kij} é igual a 1 se o vértice j é antecessor do vértice i na árvore direcionada com raiz no vértice k . As restrições (8) e (9), em conjunto com (3) e (7), garantem que cada vértice na árvore direcionada com raiz no vértice k tem no máximo um vértice antecessor, enquanto que o vértice k , pela restrição (9), não tem antecessor. As restrições (7) à (9) garantem que não existem ciclos, pois se houvesse um ciclo significaria que para algum vértice raiz k , e o conjunto de variáveis w_{kij} correspondente, existe um vértice com dois antecessores, ou um vértice raiz com um antecessor, o que não é válido segundo a definição das restrições.

4 Resultados computacionais

Nesta seção são apresentados os resultados obtidos com o método exato proposto. Primeiramente serão descritas as ferramentas utilizadas; em seguida, apresentam-se as características das instâncias criadas; e, por fim, os resultados alcançados.

4.1 Ferramentas

A ferramenta utilizada para a execução do método exato foi o CPLEX (IBM, 2013). O CPLEX é um dos pacotes de software de otimização linear mista mais utilizados na literatura. Ele é responsável por gerenciar todo o processo e, se necessário, adicionar novos cortes. Neste trabalho, todo o gerenciamento da execução do método é feito por esta ferramenta, como a realização do *Branch-and-Bound* (Land & Doig, 1960), a escolha de variáveis para o *branch* e a adição de novos

cortes. Inicialmente, os cortes adicionados são as próprias restrições do modelo, pois o algoritmo começa com o modelo vazio.

Os testes computacionais foram executados em um computador Intel Core i7 – 2600 com 3.40 GHz e 32 Gb de memória RAM, rodando um sistema operacional Arch Linux 3.9.2 – 1, que pode executar até oito *threads*. O método exato foi implementado na linguagem de programação C++, utilizando a API ILOG CPLEX 12.5.0.0, e compilado com g++ versão 4.8.0.

4.2 Instâncias

Como ainda não se tem conhecimento de trabalhos para o PAGMG-P, tivemos que criar um conjunto de instâncias para este problema. No entanto, existem instâncias para o problema geral, o PAGMG. As instâncias para o PAGMG disponíveis na literatura e utilizadas por Ferreira (2007); Ferreira et al. (2012) possuem grafos completos e simétricos. As mesmas dividem-se em dois grupos:

Grupo 1 Composto por 169 instâncias com $48 \leq |V| \leq 226$, geradas para o Problema do Caixeiro Viajante Generalizado e posteriormente introduzidas no PAGMG. Dentre as 169 instâncias, 150 possuem valor ótimo conhecido.

Grupo 2 Composto por 101 instâncias com $229 \leq |V| \leq 783$, também são adaptadas do Problema do Caixeiro Viajante. Nenhuma das 101 instâncias possui valor ótimo conhecido.

Considerando essas informações prévias, foi implementado um gerador de instâncias para o PAGMG-P. O gerador de instâncias segue, resumidamente, os seguintes passos: (i) particionar aleatoriamente o número de vértices de cada componente de maneira uniforme; (ii) definir o prêmio de cada vértice; (iii) definir o prêmio mínimo de cada componente; (iv) criar um grafo completo intracluster e definir o custo de cada aresta; e (v) criar um grafo completo intercluster e definir o custo de cada aresta. O número de vértices e o número de componentes são dados na entrada. O prêmio de cada vértice, o prêmio mínimo de cada componente e o custo de cada aresta são calculados aleatoriamente em um dado intervalo.

Foram geradas 164 instâncias com $10 \leq |V| \leq 160$ e $2 \leq m \leq 30$. Seguindo a literatura do PAGMG, todas as instâncias geradas possuem grafos completos e simétricos.

4.3 Resultados experimentais

Nesta seção, estão apresentados os resultados computacionais obtidos por meio de testes realizados com a formulação matemática proposta. Os testes foram realizados sobre 164 instâncias, apresentadas na Seção 4.2. O critério de parada estipulado foi encontrar a solução ótima ou um tempo limite de 3600 segundos (1 hora) de execução. Se a execução chegasse ao tempo limite, a melhor solução viável encontrada até o momento era retornada. O tempo de execução foi calculado considerando o tempo de todas as *threads* utilizadas. No entanto, o tempo limite foi contado considerando apenas uma *thread*. Dentre as instâncias utilizadas, foi encontrado o valor ótimo de 118, enquanto em outras 44 foi encontrada uma solução viável e nas 4 restantes o algoritmo não conseguiu uma solução viável dentro do tempo limite.

Os resultados estão descritos nas Tabelas 1 e 2. A primeira coluna das tabelas apresenta a denominação da instância. Nas três colunas seguintes têm-se o número de vértices, arestas e componentes, respectivamente. Na quinta coluna é apresentado o custo da melhor solução encontrada, e na coluna seis o tempo de execução em segundos. Essa organização se repete nas colunas seguintes. Nas tabelas, as soluções ótimas diferem das soluções viáveis por estarem em negrito. E as soluções

com valor “-” significam que o algoritmo não conseguiu encontrar uma solução viável dentro do tempo limite.

Analisando as tabelas pode-se ver que nas instâncias consideradas fáceis, em que os números de vértices e de componentes são pequenos, o algoritmo consegue encontrar a solução ótima em poucos segundos. No entanto, à medida que o número de vértices cresce, o tempo de execução aumenta, sobretudo quando o número de componentes é grande.

O algoritmo encontra solução ótima para a maioria das instâncias propostas. Porém, quando tanto o número de vértices quanto o número de componentes são grandes, o algoritmo ou não consegue encontrar uma solução ótima dentro do tempo limite, retornando a melhor solução viável encontrada, ou consegue encontrar uma solução ótima em uma quantidade de tempo muito grande.

Instância	V	E	m	Custo	T(S)	Instância	V	E	m	Custo	T(S)	Instância	V	E	m	Custo	T(S)
10a13	10	45	3	141	0.02	15a14	15	105	4	239	0.09	al20a15	20	190	5	172	0.37
25a16	25	300	6	186	2.13	30a15	30	435	5	60	2.01	30a110	30	435	10	179	2.13
35a17	35	595	7	135	3.63	40a14	40	780	4	77	8.92	40a18	40	780	8	165	32.36
40a110	40	780	10	126	6.41	45a16	45	990	6	93	16.17	45a19	45	990	9	134	43.06
45a110	45	990	10	131	141.73	45a115	45	990	15	176	13.28	50a12	50	1225	2	82	19.54
50a13	50	1225	3	135	21.2	50a15	50	1225	5	95	44.86	50a110	50	1225	10	155	70.83
50a115	50	1225	15	123	24.48	50a125	50	1225	25	157	44.24	52a18	52	1326	8	94	112.04
55a15	55	1485	5	114	214.35	55a19	55	1485	9	152	171.5	55a112	55	1485	12	133	101.13
58a14	58	1653	4	131	95.04	58a18	58	1653	8	106	119.3	58a110	58	1653	10	173	152.32
58a115	58	1653	15	188	240.43	60a12	60	1770	2	109	278.65	60a14	60	1770	4	140	175.07
60a15	60	1770	5	151	894.49	60a16	60	1770	6	147	353.51	60a110	60	1770	10	157	1510.73
60a115	60	1770	15	147	71.95	60a120	60	1770	20	145	94.96	64a14	64	2016	4	123	792.82
64a15	64	2016	5	91	3608.89	64a18	64	2016	8	147	9093.36	64a110	64	2016	10	129	1376.01
65a15	65	2080	5	104	130.07	65a110	65	2080	10	105	3056.56	65a115	65	2080	15	159	320.05
65a120	65	2080	20	-	-	65a125	65	2080	25	218	980.99	67a14	67	2211	4	72	243.09
67a110	67	2211	10	115	397.22	67a120	67	2211	20	171	867.66	69a13	69	2346	3	85	1700.46
69a19	69	2346	9	156	12951.8	69a115	69	2346	15	184	1174.23	70a14	70	2415	4	117	4381.31
70a15	70	2415	5	130	13136.7	70a17	70	2415	7	122	3255.7	70a110	70	2415	10	115	138.92
70a115	70	2415	15	166	454.11	70a118	70	2415	18	126	1035.22	70a120	70	2415	20	211	296.67
70a125	70	2415	25	127	295.57	72a13	72	2556	3	123	1666.19	72a15	72	2556	5	137	249.44
72a18	72	2556	8	111	109.13	72a110	72	2556	10	135	1512.59	72a115	72	2556	15	124	516.2
72a118	72	2556	18	165	978.21	72a120	72	2556	20	96	740.48	72a125	72	2556	25	195	1039.75
75a13	75	2775	3	92	3817.87	75a15	75	2775	5	69	114.82	75a110	75	2775	10	139	1173.7
75a115	75	2775	15	120	7282.56	75a120	75	2775	20	147	754.05	75a125	75	2775	25	157	634.72
75a130	75	2775	30	171	1671.76	78a14	78	3003	4	129	978.18	78a15	78	3003	5	124	2604.42
78a18	78	3003	8	123	1567.76	78a110	78	3003	10	141	5316.48	78a115	78	3003	15	151	1281.9
78a120	78	3003	20	170	1680.27	78a125	78	3003	25	-	-	80a12	80	3160	2	102	2517.44
80a14	80	3160	4	92	672.75	80a15	80	3160	5	99	1095.58	80a18	80	3160	8	143	2611.45
80a110	80	3160	10	134	2477.97	80a115	80	3160	15	162	2390.82	80a118	80	3160	18	154	1723.74
80a120	80	3160	20	148	4087.65	80a122	80	3160	22	155	2864.71	80a130	80	3160	30	191	5909.96
83a14	83	3403	4	142	4043.97	83a15	83	3403	5	122	12883.9	83a18	83	3403	8	138	560.63

Tabela 1: Resultados computacionais do método exato proposto.

Instância	V	E	m	Custo	T(S)	Instância	V	E	m	Custo	T(S)	Instância	V	E	m	Custo	T(S)
83a10	83	3403	10	85	9898.22	83a15	83	3403	15	149	16849.7	83a20	83	3403	20	-	-
83a25	83	3403	25	-	-	83a30	83	3403	30	181	1894.46	85a4	85	3570	4	131	1695.44
85a5	85	3570	5	111	517.84	85a8	85	3570	8	126	18351.2	85a10	85	3570	10	125	9083.65
85a15	85	3570	15	188	7298.49	85a18	85	3570	18	148	5052.42	85a22	85	3570	22	167	7832.81
85a25	85	3570	25	186	5416.95	85a30	85	3570	30	222	3674.53	90a3	90	4005	3	103	5917.89
90a4	90	4005	4	136	1618.95	90a5	90	4005	5	93	4300.88	90a8	90	4005	8	122	3787.16
90a10	90	4005	10	140	4532.56	90a15	90	4005	15	190	4820.66	90a20	90	4005	20	174	9624.66
90a25	90	4005	25	178	5927.21	90a30	90	4005	30	153	4720.42	95a3	95	4465	3	122	5866.13
95a5	95	4465	5	4504	7265.23	95a8	95	4465	8	103	2847.95	95a10	95	4465	10	139	4999.63
95a15	95	4465	15	115	6567.45	95a20	95	4465	20	202	5170.83	95a25	95	4465	25	164	6384.44
95a28	95	4465	28	150	6465.09	100a4	100	4950	4	4591	7212.56	100a5	100	4950	5	129	7581.07
100a10	100	4950	10	6115	7225.02	100a15	100	4950	15	116	6251.22	100a20	100	4950	20	4587	7175.15
100a25	100	4950	25	4762	7201.2	100a30	100	4950	30	-	-	110a5	110	5995	5	10667	7201.88
110a10	110	5995	10	6930	7256.87	110a15	110	5995	15	4164	7251.55	110a20	110	5995	20	8944	7239.04
110a25	110	5995	25	5101	7201.81	110a30	110	5995	30	10394	7670.42	120a5	120	7140	5	4678	7168.58
120a10	120	7140	10	10999	7300.03	120a15	120	7140	15	3798	7145.81	120a20	120	7140	20	11257	7201.83
120a25	120	7140	25	11757	7385.13	120a30	120	7140	30	10298	7324.23	130a5	130	8385	5	7984	7123.83
130a10	130	8385	10	6115	7109.96	130a15	130	8385	15	13327	7140.99	130a20	130	8385	20	13092	7366.28
130a24	130	8385	24	12903	7134.18	130a28	130	8385	28	12600	7659.86	140a5	140	9730	5	8118	7109.77
140a10	140	9730	10	12059	7241.54	140a15	140	9730	15	12543	7103.7	140a20	140	9730	20	14143	10976.9
140a25	140	9730	25	12549	7374.48	140a30	140	9730	30	13467	10173	150a5	150	11175	5	7222	7096.46
150a10	150	11175	10	14430	7612.63	150a15	150	11175	15	14539	7097.16	150a20	150	11175	20	14204	7122.27
150a25	150	11175	25	14266	7646.38	150a30	150	11175	30	14327	8104.3	160a5	160	12720	5	15657	7061.82
160a10	160	12720	10	14426	7449.6	160a15	160	12720	15	16096	7389.06						

Tabela 2: Resultados computacionais do método exato proposto.

5 Conclusões e trabalhos futuros

Este trabalho propôs uma nova variante para o Problema da Árvore Geradora Mínima Generalizado (PAGMG) com Prêmios nos Vértices, o PAGMG-P. O PAGMG é amplamente estudado pela literatura, mas, apesar de sua aplicação ser abrangente, esta abordagem as vezes não contempla casos particulares de alguns problemas da vida real. A variante proposta foi motivada pelas possíveis aplicações reais existentes.

Ainda não se tem conhecimento de trabalhos sobre o PAGMG-P. Logo, não foram encontradas nem formulação matemática nem instâncias para o problema na literatura. Dessa forma, este trabalho propôs uma formulação matemática e, além disso, foi implementado um gerador de instâncias, as quais foram utilizadas neste trabalho.

Os resultados quantitativos indicam que o método exato proposto se comporta de maneira eficiente para as instâncias testadas. No entanto, pode-se observar que ele apresenta resultados mais satisfatórios quando o número de vértices e componentes da instância são pequenos. Assim, se faz necessário o emprego de algoritmos mais elaborados para obtenção de melhores resultados quando estas características não são atendidas.

Como trabalhos futuros, propõe-se a implementação de um método exato mais eficaz, usando conceitos de algoritmos de planos de corte. Além disso, deve ser aplicado um pré-processamento nas instâncias, para que o número de arestas da instância possa ser reduzido. Propõe-se também a elaboração de instâncias mais difíceis, como a adaptação das instâncias do PAGMG existentes na literatura para a variante abordada.

A forma imediata de resolver problemas de otimização combinatória seria simplesmente listar todas as possíveis soluções e selecionar a melhor. No entanto, devido a complexidade e o dinamismo dos problemas práticos que, de maneira geral, possuem um grande número de soluções candidatas, esta listagem exaustiva torna-se inviável para a maioria dos casos. Por isso, o desenvolvimento de métodos que forneçam uma solução de boa qualidade, mas não necessariamente garantam ser a melhor solução existente, tem sido incentivado. Neste contexto, podem ser utilizadas as metaheurísticas. Sendo assim, propõe-se como trabalhos futuros a implementação de metaheurísticas, como o GRASP (*Greedy Randomized Adaptive Search Procedures*) e o VNS (*Variable Neighborhood Search*).

6 Agradecimentos

Agradecemos a Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES), ao Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) e a Fundação de Amparo à Pesquisa do Estado do Rio de Janeiro (FAPERJ) por financiar parcialmente este trabalho.

Referências

- Dror, M., Haouari, M. & Chaouachi, J.** (2000), 'Generalized spanning trees', *European Journal of Operational Research* **120**, 583–592.
- Feremans, C., Labbé, M. & Laporte, G.** (2000), The generalized minimum spanning tree problem: Polyhedral analysis and branch-and-cut algorithm, Technical report, Université Libre de Bruxelles, Bruxelles, Bélgica.
- Ferreira, C. M. S.** (2007), Algoritmos para o problema da árvore geradora mínima generalizado, Master's thesis, Universidade Federal Fluminense.

- Ferreira, C. S., Ochi, L. S., Parada, V. & Uchoa, E.** (2012), 'A grasp-based approach to the generalized minimum spanning tree problem', *Expert Systems with Applications* **39**, 3526–3536.
- Golden, B., Raghavan, S. & Stanojevic, D.** (2007), 'The prize-collecting generalized minimum spanning tree problem', *Journal of Heuristics* **14**, 69–93.
- IBM, I. B. M. C.** (2013), 'Ibm ilog cplex optimization studio'. URL <http://www-03.ibm.com/software/products/us/en/ibmilogcpleoptistud/>, Página visitada em 12-05-2013.
- Ihler, E., Reich, G. & Widmayer, P.** (1999), 'Class steiner trees and vlsi-design', *Discrete Applied Mathematics* **90**, 173–194.
- Kansal, A. R. & Torquato, S.** (2001), 'Globally and locally minimal weight spanning tree networks', *Physica A* **301** pp. 601–619.
- Land, A. H. & Doig, A. G.** (1960), 'An automatic method of solving discrete programming problems', *Econometrica* **28**, 497–520.
- Myung, Y. S., Lee, C. H. & Tcha, D. W.** (1995), On the generalized minimum spanning tree problem, in 'Networks 26', pp. 231–241.
- Salazar, J. J.** (1991), 'A note on the generalized steiner tree polytope', *Discrete Applied Mathematics* **100**, 137–144.
- Santos, N. M., Ochi, L. S. & Semaan, G. S.** (2012), 'Metaheuristic grasp with path-relinking to the solution of the graph partitioning problem with capacity and connectivity constraints', *Lecture Notes in Computer Science (LNCS)* **7435**, 630–641.
- Semaan, G. S., Brito, J. A. M. & Ochi, L. S.** (2010), 'Efficient algorithms for the capacity and connectivity graph partition problem', *Mecânica Computacional* **29**, 9499–9509.
- Shyu, S. J., Yin, P., Lin, B. M. T. & Haouari, M.** (2006), 'Upper and lower bounding strategies for the generalized minimum spanning tree problem', *European Journal of Operational Research* **171**, 632–647.