

O PROBLEMA DA ÁRVORE GERADORA MÍNIMA PROBABILÍSTICA HOMOGÊNEO: MODELAGEM EXATA E HEURÍSTICAS

Rafael Ferreira Barra de Souza, Sebastián Urrutia

Departamento de Ciência da Computação

Universidade Federal de Minas Gerais (UFMG)

Av. Antônio Carlos 6627 – Prédio do ICEx – Pampulha – CEP 31270-010

Belo Horizonte – Minas Gerais – Brasil

{rbarra, surrutia}@dcc.ufmg.br

RESUMO

O Problema da Árvore Geradora Mínima Probabilística é uma generalização natural do problema clássico da árvore geradora mínima e é frequentemente um modelo mais realista, que pode ser utilizado em diversas aplicações que envolvem roteamento, desenho de circuitos VLSI, logística e telecomunicações. O problema ilustra a situação pertinente quando nem todos os nós estão deterministicamente presentes, mas, eles estão presentes conforme uma determinada probabilidade. Embora existam casos específicos deste problema que podem ser tratados em tempo polinomial, sabe-se que o problema é NP-Difícil no caso geral. Neste trabalho procura-se descrever e analisar uma modelagem exata para o problema em seu caso homogêneo, isto é, quando todos os nós possuem a mesma probabilidade de estarem ativos. Adicionalmente, são apresentadas heurísticas para encontrar soluções viáveis para o problema.

PALAVRAS CHAVE. Árvore geradora mínima probabilística. Heurísticas. Otimização *a priori*.

Área principal: Teoria de grafos.

ABSTRACT

The Probabilistic Minimum Spanning Tree Problem is a natural generalization of the classical minimum spanning tree problem and is frequently a more realistic model, which can be applied to many applications that involve routing problems, VLSI design, logistics and telecommunications. The problem addresses circumstances that arise when not all nodes are deterministically present but, rather, nodes are present with known probabilities. Although there are some special cases that are solvable in polynomial time, it is known that the problem is NP-Hard in the general case. In this paper we describe and analyze an exact model that deals with the problem in its homogenous approach, where all nodes have the same probability of being active. Additionally, we present heuristics for obtaining feasible solutions to the problem.

KEYWORDS. Probabilistic Minimum Spanning Tree. Heuristics. *A priori* optimization.

Main area: Graph Theory.

1. Introdução

O problema da Árvore Geradora Mínima (MST, do inglês *Minimum Spanning Tree*) é um problema clássico de otimização combinatória em grafos, que tem como objetivo encontrar uma árvore geradora de custo mínimo em um grafo conexo onde as arestas possuem custos para conectar seus vértices. Diversos algoritmos são capazes de resolver este problema em tempo polinomial, dentre os quais é possível citar Borůvka (1926), Kruskal (1956) e Prim (1957). Tornando o problema mais realista, algumas variações foram surgindo conforme necessidades de aplicação, a exemplo da inclusão de restrições de grau, conforme Narula e Ho (1980), elementos estocásticos, a exemplo de Ishii et al. (1981), e, por fim, probabilidades individuais dos vértices estarem ativos, conforme proposição de Bertsimas (1990).

O problema da Árvore Geradora Mínima Probabilística (PMST, de *Probabilistic MST*) é modelado como uma variação do problema clássico do MST e pertence a uma família específica de problemas de otimização combinatória, composta pelos problemas de otimização *a priori*. Estes problemas, segundo Bertsimas et al. (1989), são caracterizados pela inclusão explícita de elementos probabilísticos em sua definição. Dado um grafo $G = (V, E)$, não necessariamente completo, onde há um custo associado a cada aresta e uma probabilidade associada a cada vértice, conhecidos *a priori*, deseja-se construir uma árvore geradora para este grafo. O custo ativo desta árvore é definido como a soma dos custos das arestas que conectam nós ativos em determinado momento futuro. O problema PMST tem como objetivo encontrar uma árvore geradora *a priori*, cujo custo ativo esperado seja o menor possível em um dado momento de ativação dos seus vértices, *a posteriori*.

O PMST se aplica em circunstâncias nas quais os nós estão presentes conforme uma determinada probabilidade. Mesmo que alguns casos específicos do problema possam ser resolvidos em tempo polinomial, este problema é NP-Difícil, conforme Bertsimas (1990). O problema é NP-Difícil quando o grafo não é completo e os custos das arestas são iguais, ou se o grafo é completo e os custos podem assumir dois valores distintos, 1 ou M , por exemplo. Destaca-se, ainda, a dificuldade do PMST quando a probabilidade dos vértices é idêntica para todos eles – o qual será denominado, adiante, como o caso homogêneo do PMST. Sempre que uma árvore geradora mínima possui uma topologia em estrela, o problema da árvore geradora mínima probabilística se reduz ao problema clássico da árvore geradora mínima e, então, ele pode ser resolvido em tempo polinomial. Em instâncias onde as probabilidades dos nós estarem presentes são muito baixas, conforme Bertsimas (1990), o problema é praticamente equivalente ao problema de modelagem de redes, que foi provado como sendo NP-Difícil em Garey e Johnson (1979).

A busca por uma árvore geradora de um grafo que considera que os nós possuam uma probabilidade de estarem presentes tem diversas aplicações práticas, entre as quais é possível relacionar problemas de roteamento, transporte, projetos de circuitos VLSI, estruturas organizacionais e telecomunicações. Estas aplicações são citadas em Abuali et al (1994, 1995) e exemplificadas em Bertsimas (1988a).

Existem trabalhos que abordam o PMST através de algoritmos genéticos, tal como em Abuali et al. (1994, 1995). Atualmente, conforme o trabalho de Souza e Urrutia (2008), o problema vem sendo tratado através de heurísticas construtivas baseadas no algoritmo de Prim. Neste artigo, é proposta uma formulação exata por programação linear inteira, utilizada para resolver o problema quando todas as probabilidades dos vértices são idênticas.

O restante deste artigo está organizado da maneira a seguir. Na seção 2 é apresentada a definição do PMST baseada em Bertsimas (1990). A seguir, na seção 3, descreve-se uma modelagem exata para a abordagem do problema onde os nós possuem a mesma probabilidade de estarem ativos. Na seção 4 as heurísticas de Souza e Urrutia (2008) são discutidas. Na seção 5 são descritos os resultados comparativos da modelagem exata com as heurísticas. Finalmente, na última seção, são apresentadas as conclusões e trabalhos futuros.

2. O PMST

2.1. Definição do problema

Como forma de ilustrar o problema, pode-se levar em consideração um grafo qualquer com dez nós e sua árvore geradora mínima, T , ilustrada na figura 1. A partir daí, pressupõe-se que em um momento futuro, o conjunto de nós presentes é dado por $S = \{2,3,5,6,7,10\}$. A partir deste ponto, definimos $T(S)$ como a mínima sub-árvore de T necessária para possibilitar a conexão de todos os nós de S . $T(S)$ é ilustrada na figura 2. As arestas $(4,1)$ e $(1,7)$ não estão conectadas em $T(S)$, pois os vértices 4 e 1 não estão ativos. Porém, as arestas $(6,9)$ e $(7,9)$, bem com as arestas $(5,8)$ e $(8,10)$, permanecem presentes em $T(S)$. Isto acontece pois existem nós ativos, 6 e 10, que dependem dos nós 9 e 8 para estarem conectados à sub-árvore $T(S)$.

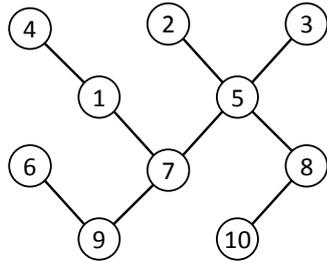


Figura 1: Árvore geradora mínima T .

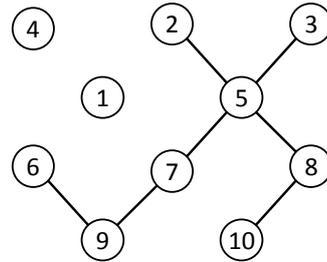


Figura 2: Sub-árvore $T(S)$.

O PMST define uma estratégia eficiente para se atualizar soluções de árvore geradora quando instâncias de um problema específico são afetadas probabilisticamente em resposta à ausência de determinados vértices do grafo. A partir de uma árvore T conhecida *a priori*, definimos $L_T(S)$ como sendo o custo da menor sub-árvore de T que conecta os nós do conjunto S . Na figura 1, se $S = \{2,3,5,6,7,10\}$, $L_T(S)$ é o custo da árvore apresentada na figura 2.

Dado um grafo não direcionado $G = (V, E)$, não necessariamente completo, uma função de custo $c: E \rightarrow \mathfrak{R}^+$, e uma função de probabilidade $P: 2^V \rightarrow [0,1]$, que mapeia cada subconjunto de nós com uma probabilidade, o objetivo do PMST é encontrar uma árvore geradora T , que minimize o custo ativo esperado $E[L_T]$, onde

$$E[L_T] = \sum_{S \subseteq V} p(S) L_T(S). \quad (1)$$

É importante notar que neste nível de generalização podemos modelar dependências entre as probabilidades de presença dos conjuntos de vértices. Outra observação adicional é que com esta formulação seria necessário um esforço de $O(n2^n)$, ($|V| = n$) operações para calcular o custo ativo esperado, $E[L_T]$, de uma árvore T .

Se um nó i está ativo com uma probabilidade p_i , sendo a ativação dos nós eventos independentes, Bertsimas (1990) mostra que o custo ativo esperado, $E[L_T]$, para uma árvore geradora mínima, é dado pela equação

$$E[L_T] = \sum_{e \in T} c(e) \left\{ 1 - \prod_{i \in K_e} (1 - p_i) \right\} \times \left\{ 1 - \prod_{i \in V - K_e} (1 - p_i) \right\}. \quad (2)$$

A operação de somatório é feita sobre todas as arestas existentes na árvore T . A remoção de uma aresta e divide T em duas sub-árvores, representadas como K_e e $V - K_e$. Na equação (2) o custo da aresta é multiplicado pela probabilidade da aresta estar ativa em um determinado momento. Pode-se calcular a probabilidade de uma aresta ser ativa como a probabilidade de se ter um nó ativo em cada uma das componentes conexas geradas pela eliminação da aresta da árvore. Isto se deve ao fato de que, em uma árvore, entre todo par de nós existe apenas um caminho.

Como as probabilidades são independentes, a probabilidade da aresta ser ativa pode ser calculada como a multiplicação das probabilidades de pelo menos um nó estar presente em cada uma das componentes conexas. A probabilidade de pelo menos um nó estar presente em cada uma das componentes conexas pode ser calculada como uma unidade menos a probabilidade de nenhum nó estar presente.

2.2. Um limite inferior para o problema

O custo ativo esperado de uma árvore T é uma função que depende do custo de suas arestas e da probabilidade de cada uma delas estar ativa. Sendo assim, $E[L_T]$ pode ser descrito a partir de uma função de probabilidade P_e , que define a probabilidade de uma aresta e estar ativa:

$$E[L_T] = \sum_{e \in T} c(e)P_e. \quad (3)$$

P_e depende exclusivamente da quantidade de nós existentes em cada sub-árvore conectada por uma aresta e , cujos tamanhos são respectivamente $|K_e| = k$ e $|V - K_e| = n - k$:

$$P_e = \{1 - (1 - p)^k\}\{1 - (1 - p)^{n-k}\}. \quad (4)$$

Considerando probabilidades idênticas para todos os nós de T , a menor probabilidade de uma aresta existir é quando um de seus nós é folha da árvore. Isto implica em $k = 1$, simplificando P_e a

$$P_e = p(1 - (1 - p)^{n-1}). \quad (5)$$

Supondo uma árvore onde cada aresta está presente conforme sua menor probabilidade possível, *i.e.* uma árvore de topologia em estrela, o custo ativo esperado pode ser limitado inferiormente conforme

$$\sum_{e \in T} c(e)P_e \geq \sum_{e \in T} c(e)p(1 - (1 - p)^{n-1}). \quad (6)$$

Levando-se em conta que P_e é constante, sendo possível retirá-lo do somatório, e, sabendo-se que o limite inferior sobre o custo $\sum_{e \in T} c(e)$ de qualquer árvore geradora é o custo de sua árvore geradora mínima (o qual chamaremos L_{AGM}) outro limite inferior pode ser determinado para uma árvore T :

$$p(1 - (1 - p)^{n-1}) \sum_{e \in T} c(e) \geq p(1 - (1 - p)^{n-1})L_{AGM} \quad (7)$$

Sendo assim, tem-se um limite inferior para o PMST, cujo custo ativo esperado é

$$E[L_T] = p(1 - (1 - p)^{n-1})L_{AGM}. \quad (8)$$

3. Formulação exata

Nesta seção é proposta uma modelagem em programação linear inteira que pode ser utilizada para resolver o problema em sua abordagem homogênea, onde todos os nós possuem a mesma probabilidade, ou seja, $p_i = p$, tendo-se $0 < p < 1$.

Para este problema, tem-se variáveis de decisão x . Assumem valor 1 se existe uma aresta (i, j) tal que, quando removida, divide a árvore em duas sub-árvores de k e $n - k$ nós, e 0 caso contrário. De maneira auxiliar recorre-se ao uso das variáveis y , que determinam o valor de k para uma aresta (i, j) .

$$x_{(i,j)k} \in \{0,1\} \quad \forall (i,j) \in E; k = 1,2, \dots, \frac{n}{2}; i < j \quad (9)$$

$$y_{ij} \in \mathbb{Z}_+ \quad \forall (i,j) \in E; i < j \quad (10)$$

$$s_{ij} \in \{0,1\} \quad \forall (i,j) \in E; i < j \quad (11)$$

$$Pc_{(i,j)k} = c_{ij} \cdot (1 - q^k)(1 - q^{n-k}), \quad q = 1 - p \quad (12)$$

A função objetivo do problema é:

$$\min \sum_{(i,j) \in E} \sum_{k=1}^{\frac{n}{2}} Pc_{(i,j)k} x_{(i,j)k} \quad (13)$$

Tendo como restrições:

$$\sum_{(i,j) \in E} \sum_{k=1}^{\frac{n}{2}} x_{(i,j)k} = n - 1 \quad (14)$$

$$\sum_{k=1}^{\frac{n}{2}} x_{(i,j)k} \leq 1, \quad \forall (i,j) \in E, i < j \quad (15)$$

$$\sum_{k=1}^{\frac{n}{2}} kx_{(i,j)k} = y_{ij}, \quad \forall (i,j) \in E, i < j \quad (16)$$

$$y_{ij} + n(1 - s_{ij}) + n \left(1 - \sum_{k=1}^{\frac{n}{2}} x_{ijk} \right) \geq 1 + \sum_{\substack{(i,l) \in E \\ l \neq j}} y_{il}, \quad \forall (i,j) \in E; i < j \quad (17)$$

$$y_{ij} + n \cdot s_{ij} + n \left(1 - \sum_{k=1}^{\frac{n}{2}} x_{ijk} \right) \geq 1 + \sum_{\substack{(j,l) \in E \\ l \neq i}} y_{jl}, \quad \forall (i,j) \in E; i < j \quad (18)$$

A figura 3 é importante no auxílio do entendimento da formulação proposta. Nela é exemplificada a remoção de uma aresta de custo $c(e)$ que conecta os vértices 7 e 5 de uma árvore específica, dividindo-a em duas sub-árvores representadas pelo conjunto de vértices k e seu complemento $n - k$, sendo n o total de vértices da árvore.

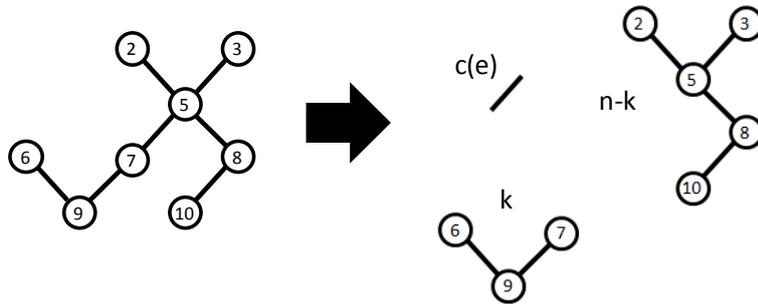


Figura 3: Exemplo de remoção de aresta (5,7) de um grafo de exemplo.

A função objetivo (13) tem como propósito minimizar o custo ativo esperado $E[L_T]$ da árvore geradora mínima probabilística, calculado a partir do somatório das funções de custo de probabilidade, aqui chamadas de Pc (do inglês *Probability Cost*), multiplicado pela variável de decisão $x_{(i,j)k}$. $x_{(i,j)k}$ representa a existência de uma aresta (i,j) que realize a conexão de duas sub-árvores de tamanho igual a k e $n - k$. Considera-se que k atinge no máximo $\frac{n}{2}$ já que, por simetria, este é o maior corte possível em uma árvore de n vértices. Cortes com $n > k > \frac{n}{2}$ produzem sub-árvores com os mesmos tamanhos, apenas invertendo qual das sub-árvores possui mais ou menos nós.

A inclusão da função Pc na função objetivo, em comparação à figura 3, insere a função de probabilidade apresentada em (1) e expandida em (2) para o cálculo do custo ativo da árvore, que depende da probabilidade dos nós presentes em cada uma das sub-árvores geradas pela remoção da aresta em questão. Uma aresta terá custo ativo alto à medida que conectar duas sub-árvores que tenham grande probabilidade de ter pelo menos um nó ativo. Uma aresta terá custo ativo baixo caso conecte sub-árvores com baixa probabilidade de pelo menos um nó estar ativo, o que torna o valor da função Pc mais baixo. Quanto mais altos forem os valores das probabilidades dos vértices estarem ativos, maior será o valor atribuído a Pc . Por questões de simplificação em (12), a probabilidade de um nó estar ativo, denotada por p , é convenientemente substituída por q , que se trata da probabilidade de um nó não estar ativo em momentos de ativação.

Discutindo-se as restrições, (14) garante que o número de arestas seja igual a $n - 1$, quantidade de arestas necessária para a construção de uma árvore geradora. A restrição (15) informa que uma determinada aresta pode ou não ser usada, de forma que, se usada, ela indica um determinado valor de k . As restrições agrupadas em (16) realizam a criação de uma variável y que indica o valor do corte, ou seja, o valor de k de uma aresta (i, j) da árvore.

As restrições (17) e (18) calculam o valor de y para cada aresta (i, j) . Este cálculo é feito a partir dos valores das variáveis y para as arestas incidentes no vértice i (restrição 17) e a partir dos valores das variáveis y para as arestas incidentes no vértice j (restrição 18). O segundo termo da parte esquerda das inequações desativa (torna trivialmente satisfeita) uma das duas restrições conforme o valor da variável s_{ij} . O terceiro termo da parte esquerda desativa as restrições quando a aresta (i, j) não faz parte da árvore. Portanto, apenas uma das restrições estará ativa para cada aresta da árvore. Estando uma das restrições ativa, o seu significado é que o valor de y de uma aresta (i, j) deve ser maior ou igual à soma do valor de y para todas as arestas incidentes ao mesmo vértice da aresta em consideração, acrescida de uma unidade.

Com relação à quantidade de variáveis e restrições, sendo E o conjunto de arestas do grafo utilizado e V o seu conjunto de vértices, a modelagem proposta apresenta $O(|V| \times |E|)$ variáveis x , $O(|E|)$ variáveis y , $O(|E|)$ variáveis s e, por fim, $O(|E|)$ restrições.

Essa modelagem só é possível graças à consideração da probabilidade de todos os nós ser a mesma. Em uma abordagem heterogênea, onde os nós possuam probabilidades diferentes exige uma modelagem bem mais complexa – e desconhecida até então – já que o grande fator que possibilita a modelagem exibida neste momento é que basta que se saiba o número de nós existentes de cada lado de uma aresta para saber a probabilidade de existência de um nó ativo em cada sub-árvore.

4. Heurísticas

Até o presente momento não existe na literatura qualquer abordagem para o problema, senão através de heurísticas. Em Abuali et al. (1994, 1995) é estudado o caso homogêneo do problema com instâncias de até 20 nós. O trabalho de Souza e Urrutia (2008) estuda os casos homogêneo e heterogêneo, com instâncias de até aproximadamente 2400 nós. Como forma de se encontrar soluções viáveis para o PMST, dada a complexidade de sua resolução, foram utilizadas heurísticas construtivas propostas em Souza e Urrutia (2008). O algoritmo de Prim (1957), descrito no algoritmo 1 em sua forma clássica, serve como base para a elaboração das heurísticas.

Algoritmo 1: Algoritmo de Prim	
INÍCIO PRIM(Grafo G)	
1.	Iniciar T com um vértice qualquer de G
2.	Manter um conjunto contendo as arestas de G
3.	enquanto (os vértices de G não foram todos conectados)
4.	Ligar a T o vértice que se conecta a T com menor custo
5.	Remover a aresta correspondente do conjunto de arestas
6.	fim enquanto
7.	A árvore T resultante é uma árvore geradora mínima
8.	FIM PRIM(Árvore T)

A primeira heurística proposta constrói a árvore geradora por meio do algoritmo de Prim, sem considerar as probabilidades dos nós. O custo de inserção de uma aresta depende apenas do custo individual das arestas. A segunda heurística, chamada de Prim-D, realiza o cálculo do custo de inserção de cada aresta a ser conectada baseando-se no caminho mais longo do vértice ao qual o candidato a ser inserido se conecta para outro nó da árvore. Mantendo os caminhos mais longos menores, é esperado que as probabilidades das arestas sejam menores. A terceira heurística, chamada Prim-P, realiza a computação do custo das arestas através do custo original da aresta no grafo, dividido pela probabilidade do vértice estar presente. No algoritmo de Prim, o quanto antes um vértice é escolhido para ser conectado à árvore, maiores são as chances de que este seja um nó essencial para a conectividade da árvore, dado que nos seguintes passos vários outros nós podem se conectar a ele. Observa-se que o último nó inserido pelo algoritmo é sempre uma folha da árvore. Esta última heurística, portanto, se baseia na expectativa de que os vértices que têm mais chances de estarem ativos sejam nós internos da árvore, ao contrário dos vértices com probabilidade pequena. Como os nós mais internos são aqueles que mantêm a conectividade da árvore, é desejável que estes tenham sempre uma probabilidade maior de estarem ativos – caso contrário, em uma situação real com vértices com probabilidade pequena de estarem ativos, ao se localizarem mais internamente na árvore, devem ser tornados ativos meramente para garantir a conectividade de todos os vértices ativos.

5. Resultados

5.1. Ambiente e instâncias de testes

A leitura das instâncias e a codificação das heurísticas utilizadas neste trabalho foram desenvolvidas na linguagem C++ e executadas em uma máquina com processador Intel Core 2 Duo de 1.83GHz, 3GB de RAM e sistema operacional Windows Vista SP1. O modelo de programação linear inteira foi implementado no *solver* iLOG CPLEX 10.2.0 e resolvido na mesma máquina.

Para a realização dos testes foram utilizadas instâncias correspondentes a grafos completos, disponíveis para o *Challenge Traveling Tournament Problem*. Considerou-se que todos os nós possuíam a mesma probabilidade de estarem ativos, sendo realizados testes para probabilidades iguais a 0.1 e 0.5.

Os testes comparam os custos ativos esperados das árvores geradas através das heurísticas aos daquelas geradas pela modelagem em programação linear inteira. O tempo gasto para a construção das árvores conforme cada uma das ferramentas, instâncias e probabilidades também foi amostrado e comparado. Para algumas instâncias, o tempo de execução do *solver* foi limitado em 22000 segundos – aproximadamente 6h – e o *gap* entre os limites superior e inferior foi coletado.

5.2. Comparação dos resultados

Na comparação do modelo de programação linear inteira com as heurísticas, foi considerado que todos os vértices do grafo possuíam a mesma probabilidade. O custo das árvores geradas em cada caso foi calculado conforme a equação (2) para o custo ativo esperado $E[L_T]$. Foram comparados os custos ativos das árvores geradas entre as heurísticas e o modelo, além de ter sido feita uma comparação com um limite inferior para o problema. A intenção de se utilizar um limite inferior nesta comparação é de verificar a distância dos resultados das heurísticas para um limite inferior, bem como a distância dos resultados da modelagem exata nos casos em que a execução do *solver* foi interrompida segundo um critério de tempo.

Avaliando os resultados da tabela 1, para probabilidades iguais a 0.5, é possível notar que para instâncias pequenas as heurísticas encontraram custo ativo muito próximo ao valor ótimo, encontrado pela modelagem exata. Em alguns casos, conseguiu-se até encontrar o valor ótimo com as heurísticas. Mesmo que as instâncias sejam maiores e o resultado das heurísticas seja mais distante percentualmente do limite inferior, este aspecto não parece indicar piora na

qualidade das heurísticas conforme as instâncias, já que a distância entre o valor ótimo e o limite inferior também aumenta conforme a quantidade de vértices das instâncias aumenta.

Quando se analisa a tabela 1 junto à tabela 3, que mede os tempos de execução das heurísticas e do *solver* para cada instância, nota-se um aspecto favorável às heurísticas. Para instâncias de 12 a 16 nós, que tiveram a execução do *solver* interrompida após 22000 segundos, as soluções informadas pelo *solver* foram muito próximas das soluções fornecidas pelas heurísticas. Pode-se dizer que em comparação à modelagem exata, as heurísticas são capazes de informar soluções viáveis razoáveis em pouco tempo, o que pode ser de grande utilidade em alguns cenários. Ainda nas situações em que a execução do *solver* foi limitada ao tempo de execução, percebe-se que o *gap* entre os limites superior e inferior aumenta, ao passo que as heurísticas informam soluções de custo menor.

Tabela 1: Comparação dos resultados exatos com as heurísticas para probabilidade de 0.5

Instâncias		Limite Inferior	Modelagem exata			Prim Clássico / Prim-P			Prim-D		
		$p=[0.5]$	$p=[0.5]$			$p=[0.5]$			$p=[0.5]$		
<i>instância</i>	<i>nós</i>	<i>val.</i>	<i>custo</i>	<i>gap(%)</i>	<i>L.Inf.(%)</i>	<i>custo</i>	<i>L.Inf.(%)</i>	<i>Opt.(%)</i>	<i>custo</i>	<i>L.Inf.(%)</i>	<i>Opt.(%)</i>
<i>inst4</i>	4	473,38	483,38	0,00	2,11	483,38	2,11	0,00%	483,38	2,11	0,00%
<i>inst6</i>	6	871,88	1075,60	0,00	23,37	1157,83	32,80	7,65%	1075,62	23,37	0,00%
<i>inst8</i>	8	1067,59	1351,40	0,00	26,58	1448,80	35,71	7,21%	1363,78	27,74	0,92%
<i>inst10</i>	10	1246,56	1340,95	0,00	7,57	1736,44	39,30	29,49%	1652,56	32,57	23,24%
<i>inst12</i>	12	1984,53	2490,31	22,05	25,49	2598,70	30,95	4,35%	2515,11	26,74	1,00%
<i>inst14</i>	14	2630,18	3841,68	39,43	46,06	3697,26	40,57	-3,76%	3697,26	40,57	-3,76%
<i>inst16</i>	16	2657,42	4352,60	46,83	63,79	4078,60	53,48	-6,30%	3995,09	50,34	-8,21%

Legenda: *val.*: valor calculado para o limite inferior da instância; *custo*: custo da árvore geradora; *gap*: distância entre os limites inferior e superior conforme o *solver*; *L.Inf.*: percentual do *custo* da árvore geradora com relação ao limite inferior (*val.*); *Opt.*: percentual do *custo* (heurística) com relação ao *custo* da solução da modelagem.

A tabela 2, que contém os resultados para instâncias com probabilidades iguais a 0.1, é qualitativamente análoga à tabela 1, porém é perceptível o aumento da dificuldade em se resolver o problema quando se tem probabilidades muito próximas de 0. À medida que $p \rightarrow 0$, o problema se torna mais difícil, aproximando-o do problema de modelagem em redes, conforme Bertsimas (1988a). Por outro lado, assim como nos resultados para a probabilidade igual a 0.5, as heurísticas demonstraram soluções até de custo menor (para instâncias maiores) do que as do *solver*, além do tempo ter sido muito menor. Quanto menor a probabilidade dos nós, mais interessante pode se tornar o uso das heurísticas – e maior custo computacional é necessário na resolução exata do problema.

Tabela 2: Comparação dos resultados exatos com as heurísticas para probabilidade de 0.1

Instâncias		Limite Inferior	Modelagem exata			Prim Clássico / Prim-P			Prim-D		
		$p=[0.1]$	$p=[0.1]$			$p=[0.1]$			$p=[0.1]$		
<i>instância</i>	<i>nós</i>	<i>val.</i>	<i>custo</i>	<i>gap(%)</i>	<i>L.Inf.(%)</i>	<i>custo</i>	<i>L.Inf.(%)</i>	<i>Opt.(%)</i>	<i>custo</i>	<i>L.Inf.(%)</i>	<i>Opt.(%)</i>
<i>inst4</i>	4	29,32	30,04	0,00	2,45	30,04	2,46	0,00%	30,04	2,46	0,00%
<i>inst6</i>	6	73,71	95,60	0,00	29,69	100,60	36,47	5,23%	96,72	31,21	1,18%
<i>inst8</i>	8	112,27	155,81	0,00	38,78	165,42	47,34	6,17%	160,69	43,13	3,13%
<i>inst10</i>	10	153,02	226,94	24,87	48,31	244,23	59,61	7,62%	240,04	56,87	5,77%
<i>inst12</i>	12	272,49	391,69	35,96	43,75	419,85	54,08	7,19%	416,09	52,70	6,23%
<i>inst14</i>	14	392,37	673,16	48,76	71,56	725,03	84,78	7,71%	721,63	83,91	7,20%
<i>inst16</i>	16	422,07	1056,61	65,42	150,34	954,10	126,05	-9,70%	950,98	125,31	-10,00%

Legenda: *val.*: valor calculado para o limite inferior da instância; *custo*: custo da árvore geradora; *gap*: distância entre os limites inferior e superior conforme o *solver*; *L.Inf.*: percentual do *custo* da árvore geradora com relação ao limite inferior (*val.*); *Opt.*: percentual do *custo* (heurística) com relação ao *custo* da solução da modelagem.

A dificuldade na resolução do problema para probabilidades muito baixas é reforçada através de uma análise da tabela 3, que compara as heurísticas e a modelagem exata com relação ao tempo de execução. O tempo de execução para instâncias onde as probabilidades são iguais a 0.1 é sempre maior que as instâncias de probabilidades iguais a 0.5. Isto recorre a um aspecto

importante: como no caso de probabilidades bem baixas o problema se aproxima do problema de modelagem em redes, talvez fosse interessante utilizar uma modelagem em programação linear inteira do problema de modelagem de redes e comparar à modelagem proposta para o PMST. Por mais que probabilidades de valor pequeno tenham aumentado consideravelmente o tempo de execução do *solver*, o uso de uma modelagem exata para probabilidade maior, tal como 0.5, apresentou tempos de execução proibitivos para instâncias com mais de 8 vértices. Mais uma vez, o uso de heurísticas se justifica, pois nelas o tempo de execução foi praticamente instantâneo.

Entre as heurísticas, uma atenção especial ao caminho mais longo dentro da árvore acaba se tornando uma estratégia interessante na abordagem homogênea do PMST. Os resultados da heurística Prim-D foram ligeiramente melhores em termos do custo ativo da árvore gerada para ambas probabilidades avaliadas.

Tabela 3: Comparação do tempo de execução do *solver* e das heurísticas

Instâncias		Modelagem exata	Prim Clássico / Prim-P / Prim-D	Modelagem exata	Prim Clássico / Prim-P / Prim-D
		$p=[0.5]$	$p=[0.5]$	$p=[0.1]$	$p=[0.1]$
<i>instância</i>	<i>nós</i>	<i>tempo(s)</i>	<i>tempo(s)</i>	<i>tempo(s)</i>	<i>tempo(s)</i>
<i>inst4</i>	4	2,350	0,015	2,570	0,012
<i>inst6</i>	6	6,640	0,019	7,350	0,020
<i>inst8</i>	8	68,172	0,027	232,890	0,026
<i>inst10</i>	10	2366,220	0,032	22000+	0,029
<i>inst12</i>	12	22000+	0,033	22000+	0,036
<i>inst14</i>	14	22000+	0,040	22000+	0,040
<i>inst16</i>	16	22000+	0,052	22000+	0,049

6. Conclusões

Este trabalho apresentou um modelo de programação inteira para o problema da árvore geradora mínima probabilística, considerando a abordagem em que todos os nós possuem a mesma probabilidade de estarem ativos. O problema é NP-Difícil e sua aplicação a situações reais resulta em instâncias de grande porte. Para solucioná-las em tempo hábil, torna-se necessária a utilização de heurísticas, cuja utilidade é ainda mais evidente à medida que se utiliza instâncias maiores e tem-se probabilidades menores de os nós estarem ativos. Neste trabalho reforça-se a justificativa do uso de heurísticas para trabalhar com o problema, até presente em Souza e Urrutia (2008). Poderia-se tentar melhorar a modelagem exata de duas formas. A primeira delas seria tentar melhorar as duas últimas restrições, que tratam dos cortes na árvore geradora e prevenção de ciclos. A segunda seria tentar utilizar a formulação do problema de modelagem de redes para tentar reduzir o tempo de execução do modelo exato no *solver* para instâncias de probabilidades muito pequenas. De qualquer forma, para o caso onde as probabilidades dos vértices são distribuídas entre 0.1 e 0.9, a direção a seguir é na utilização de heurísticas – permanecendo sem abordagem exata.

Trabalhos futuros incluem o uso de metaheurísticas e algoritmos de busca local. Aproveitando de técnicas de reotimização e cálculo do custo ativo esperado em $O(n)$, conforme proposições em Bertsimas (1988b) para a abordagem homogênea do problema, é possível que se consiga melhorar ainda mais o tempo de execução das heurísticas, auxiliando no desenvolvimento de uma técnica de busca local eficiente para o PMST. Pretende-se também verificar o comportamento das heurísticas em outros tipos de instâncias, principalmente procurando-se refinar a heurística Prim-D, usufruindo-se assim de sua atuação mais eficiente em instâncias onde as probabilidades dos nós são iguais.

7. Agradecimentos

Os autores agradecem ao CNPq pelo financiamento parcial de seus trabalhos através do projeto 55.0790/2007-1 - CNPq/CTINFO/OTIMAL II.

Referências

- Abuali, F. N.; Schoenefeld, D. A.; Wainwright, R. L.**, (1994). *Designing Telecommunications Networks Using Genetic Algorithms and Probabilistic Minimum Spanning Trees*, Proceedings of the 1994 ACM Symposium on Applied Computing, Phoenix, ACM Press, p. 242-246.
- Abuali, F. N.; Wainwright, R. L.; Schoenefeld, D. A.**, (1995). *Determinant Factorization: A New Encoding Scheme for Spanning Trees Applied to the Probabilistic Minimum Spanning Tree Problem*, Proceedings of The 6th Intl. Conf. on Genetic Algorithms (ICGA-95), Pittsburgh, PA. Morgan Kaufmann Publisher, p. 470-477.
- Bertsimas, D. J.**, (1988a). *The Probabilistic Minimum Spanning Tree, Part I: Complexity and Combinatorial Properties*, Massachusetts Institute of Technology, Sloan School of Management Working Paper, Rm E53-359.
- Bertsimas, D. J.**, (1988b). *The Probabilistic Minimum Spanning Tree, Part II: Probabilistic Analysis and Asymptotic Results*, Massachusetts Institute of Technology, Sloan School of Management Working Paper, OR184-88.
- Bertsimas, D. J., Jaillet P., Odoni, A. R.** (1989) *A Priori Optimization*, Massachusetts Institute of Technology, Sloan School of Management Working Paper No. 3059-89-MS – August 1989.
- Bertsimas, D. J.**, (1990). *The Probabilistic Minimum Spanning Tree Problem*, Networks, Volume 20, p. 245-275.
- Borůvka, O.** (1926). *O jistém problému minimálním*, Práce mor. přírodověd. spol. v Brně III 3, p. 37–58.
- Challenge Traveling Tournament Instances** (n.d.). *Challenge Traveling Tournament Instances, a library of sample instances for the TTP* – Disponível em: <http://mat.gsia.cmu.edu/TOURN/>, consultado pela última vez em 12 de julho de 2009).
- Garey, M. R., Johnson, D. S.** (1979). *Computers and Intractability: A Guide to the Theory of NPCompleteness*. Freeman, New York.
- Ishii, H., Shiode, H., Nishida, T.** (1981). *Stochastic spanning tree problem*, Discrete Applied Mathematics 3, p. 263-273.
- Kruskal, J.B. Jr.**, (1956). *On the shortest spanning subtree of a graph and the traveling salesman problem*, Proc. ACM 7/1, p. 48-50.
- Narula, S.C., Ho, C.A.** (1980). *Degree-constrained minimum spanning tree*, Computer & Operations Research 7, p. 239-249.
- Prim, R.C.** (1957). *Shortest connection networks and some generalizations*, Bell System Technical Journal 36, p. 1389-1401.
- Souza, R. F. B., Urrutia, S. (2008).** *Heurísticas para o problema da Árvore Geradora Mínima Probabilística*, Anais do XL Simpósio Brasileiro de Pesquisa Operacional, p. 1344-1353.