

# O Problema da Árvore Geradora Mínima Fuzzy: um algoritmo para o caso envolvendo incertezas nos pesos das arestas e na estrutura da rede

Fábio Hernandes<sup>1</sup>, Cassiano Blonski Sampaio<sup>1,2</sup>

<sup>1</sup> Departamento de Ciência da Computação, Universidade Estadual do Centro-Oeste, DECOMP/UNICENTRO-PR, 85.015-430, caixa postal 3010, Guarapuava-PR  
[hernandes@unicentro.br](mailto:hernandes@unicentro.br), [cassianobsampaio@gmail.com](mailto:cassianobsampaio@gmail.com)

<sup>2</sup> Bolsista CNPq - Iniciação Científica

**Resumo.** O problema da árvore geradora mínima com incertezas nos parâmetros/estruturas aparece em diversas aplicações, tais como: instalações de linhas telefônicas e elétricas, armazenamento de informações, transportes, problemas de cluster, dentre outros. Com isso, neste trabalho é proposto um algoritmo que aborda o problema da árvore geradora mínima considerando incertezas nos pesos das arestas, bem como na estrutura da rede em si, sendo essas incertezas modeladas por meio da teoria dos conjuntos *fuzzy* e utilizados os conceitos de dominância para eliminar as arestas com pesos maiores. Duas instâncias são executadas para ilustrar o comportamento do algoritmo proposto.

**Keywords:** Teoria dos Grafos, Algoritmo de Prim, Teoria dos Conjuntos *Fuzzy*.

## 1 Introdução

O Problema da Árvore Geradora Mínima, PAGM, consiste em, a partir de um grafo não orientado, conexo e com pesos em cada aresta, gerar uma árvore que conecte todos os vértices do grafo e que tenha o menor peso total. Este problema possui diversas aplicações na Engenharia e na Informática, dentre as quais se destacam [1,2,3]: instalações de linhas telefônicas (ou elétricas) entre um conjunto de localidades utilizando a infraestrutura das rodovias com o menor uso de material, análise de clusters, armazenamento de informações, transportes, dentre outras. Os principais algoritmos da literatura que resolvem este problema são os algoritmos clássicos de Kruskal, Prim, Borůvka, e Sollin [4].

Em algumas aplicações do PAGM os parâmetros da rede ou a estrutura em si não são exatos, pois, em certas situações, estes representam, por exemplo, tempo, custos e conexões [5,6]. Em alguns trabalhos o PAGM com incertezas é abordado por meio da teoria dos conjuntos *fuzzy* [7,8,9], ferramenta frequentemente utilizada para dar um tratamento matemático a termos linguísticos como “aproximadamente” e “em torno de”, sendo assim denominado de *Problema da Árvore Geradora Mínima Fuzzy – PAGM-Fuzzy* [10].

Na literatura há alguns trabalhos que tratam do PAGM-*Fuzzy*, dentre os quais se destacam: Delgado *et al.* [10], Chunde [1], Takahashi e Yamakami [5,6], Janiak e Kasperski [11], Mohanty *et al.* [12], Nayeem e Pal [13], Dhanalakshmi e Parvathi [14], Hernandez e Lourenço [15], Gao [16], Theresa e Raj [17], Zhang e Zhang [18] e Zhou *et al.* [19]. No trabalho de Delgado *et al.* [10] foi introduzido o conceito da árvore geradora mínima *fuzzy*. Já em Chunde [1] foi proposto um algoritmo para o PAGM-*Fuzzy* com incertezas nas estruturas, sendo uma adaptação de Delgado *et al.* [10]. No primeiro trabalho de Takahashi e Yamakami [5] foi tratado do problema com parâmetros *fuzzy*, resolvendo-o por meio de algoritmos genéticos, enquanto que no segundo trabalho [6] abordaram o problema em uma rede com estrutura incerta, sugerindo um algoritmo baseado nos trabalhos de Delgado *et al.* [10] e Chunde [1]. Janiak e Kasperski [11] utilizaram a teoria de possibilidade de Zadeh [9] nos pesos das arestas e resolveram o problema por meio de um algoritmo clássico da literatura. Mohanty *et al.* [12] aplicaram o índice de ordenação de Yao e Lin [20] nos parâmetros e resolveram o problema por meio do algoritmo clássico de Borůvka. Nayeem e Pal [13] estudaram o problema com parâmetros *fuzzy*, resolvendo-o por meio de algoritmos genéticos, assim como Takahashi e Yamakami [5]. Dhanalakshmi e Parvathi [14] trataram do problema da árvore geradora mínima, sendo os pesos das arestas abordados como números *fuzzy* trapezoidais, propondo um algoritmo que utiliza o centroide e o resolve como um problema *crisp*. Hernandez e Lourenço [15] resolveram o problema da árvore geradora mínima com estrutura incerta, propondo um algoritmo exato, baseado em Takahashi e Yamakami [6], que apresenta um conjunto solução de árvores geradoras mínimas. Gao [16] abordou o conceito de árvore *fuzzy*, propondo o índice de ciclo incerto e de caminho incerto, utilizando matrizes de adjacência. Theresa e Raj [17] abordaram o problema da árvore geradora, apresentando um algoritmo híbrido que utiliza redes neurais artificiais e lógica *fuzzy*, aplicando-o ao problema real de classificação dos casos de assassinatos. Zhang e Zhang [18] aplicaram o problema da árvore geradora mínima em redes veiculares *ad hoc*, por meio de um algoritmo de colônia de abelhas. Em um dos mais recentes trabalhos, Zhou *et al.* [19] introduziram o conceito de  $\alpha$ -árvore geradora mínima, utilizando a definição de credibilidade e resolvendo o problema via algoritmos de Kruskal e Prim.

Analisando os trabalhos da literatura descritos anteriormente, verifica-se que estes resolvem o problema com incertezas nos parâmetros ou na estrutura da rede, não abordando casos em que estas ocorram concomitantemente. Além do mais, estes resolvem o problema via algoritmos genéticos ou pelos algoritmos clássicos da literatura (problemas *crisp*). Com a finalidade de abordar o PAGM-*Fuzzy* de forma distinta à literatura, neste trabalho é abordado o problema da árvore geradora mínima com incertezas nos pesos das arestas (parâmetros) e na estrutura da rede em si, propondo um algoritmo que mantenha as incertezas até a obtenção da solução final.

Este trabalho está organizado da seguinte maneira: a Seção 2 contém conceitos e definições do problema da árvore geradora mínima e da teoria dos conjuntos *fuzzy*, a Seção 3 apresenta o algoritmo proposto, nas Seções 4 e 5 estão os resultados computacionais, as conclusões e trabalhos futuros.

## 2 Conceitos e Definições

Nesta seção são descritos os problemas da árvore geradora mínima *crisp* e *fuzzy*, os conceitos da teoria dos conjuntos *fuzzy* utilizados no trabalho, além da definição da relação de ordem aplicada para eliminar as árvores com pesos maiores.

### 2.1 O problema da árvore geradora mínima

**Definição 1.** [21] Um grafo não orientado  $G=(N,A)$ , tal que  $N$  é o conjunto dos vértices e  $A$  o das arestas, é uma *árvore* se, e somente se,  $G$  for conexo e acíclico.

Seja um grafo  $G=(N,A)$ , tal que para cada aresta  $(i,j) \in A$  associa-se um número  $C_{ij}$ , denominado *peso da aresta*  $(i,j)$ , e dado um subgrafo  $G'=(N,A')$  de  $G=(N,A)$ , chama-se *peso de  $G'$*  o valor  $C(G')$  calculado da seguinte forma:

$$C(G') = \sum_{(i,j) \in A'} C_{ij}$$

sendo o grafo  $G=(N,A)$  conexo. Caso o grafo seja desconexo, o problema pode ser resolvido separadamente para cada componente conexo.

**Definição 2.** [21] O *problema da árvore geradora mínima* consiste em procurar uma árvore  $T^*$  de  $G$ , tal que:

$$C(T^*) = \min_T \{C(T)\}$$

sendo o mínimo considerado sobre o conjunto de todas as possíveis árvores  $T$  de  $G$ .

Dado um grafo conexo  $\tilde{G}=(N,A)$ , tal que para cada aresta  $(i,j) \in A$  associa-se um número *fuzzy*  $\tilde{C}_{ij}$  – chamado *peso da aresta*  $(i,j)$  – e seja  $\tilde{G}'=(N,A')$  um subgrafo de  $\tilde{G}=(N,A)$ , denota-se *peso de  $\tilde{G}'$*  o valor:

$$\tilde{C}(\tilde{G}') = \sum_{(i,j) \in A'} \tilde{C}_{ij}$$

**Definição 3.** O *problema da árvore mínima fuzzy* consiste em procurar uma árvore  $T^*$  de  $\tilde{G}=(N,A)$ , tal que:

$$\tilde{C}(T^*) = \min_T \{\tilde{C}(T)\}$$

sendo o mínimo considerado sobre o conjunto de todas as possíveis árvores de  $\tilde{G}$ .

### 2.2 Teoria dos conjuntos *fuzzy*

A teoria dos conjuntos *fuzzy* foi introduzida por Zadeh [7,8,9] com o intuito de dar um tratamento matemático a certos termos linguísticos subjetivos, sendo uma ferramenta frequentemente utilizada na modelagem de incertezas. A seguir, são apresentadas as definições utilizadas neste trabalho [22,23].

**Definição 4.** Um *número fuzzy triangular*, denotado por  $\tilde{a}=(m-\delta, m, m+\beta)$ , tem sua *função de pertinência*  $\mathfrak{R} \rightarrow [0,1]$ ,  $\mu_{\tilde{a}}(x)$ , definida pela expressão:

$$\mu_{\tilde{a}}(x) = \begin{cases} 0, & \text{se } x < (m - \delta) \\ \frac{x - (m - \delta)}{\delta}, & \text{se } (m - \delta) \leq x \leq m \\ \frac{(m + \beta) - x}{\beta}, & \text{se } m < x \leq (m + \beta) \\ 0, & \text{se } x > (m + \beta) \end{cases}$$

tal que:  $m$  é o valor modal;  $\delta$  o espalhamento à esquerda e  $\beta$  o espalhamento à direita.

**Definição 5.** Valor modal é o valor  $x \in [m - \delta, m + \beta]$  para o qual a função de pertinência tem valor máximo.

**Definição 6.** Sejam  $\tilde{a}$  e  $\tilde{b}$  dois números *fuzzy* triangulares,  $\tilde{a} = (m_1 - \delta_1, m_1, m_1 + \beta_1)$  e  $\tilde{b} = (m_2 - \delta_2, m_2, m_2 + \beta_2)$ , a soma *fuzzy* é denotada por:

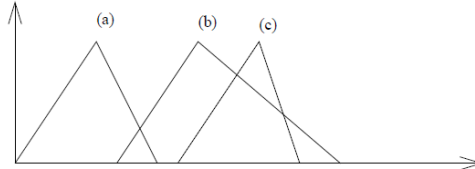
$$\begin{aligned} \tilde{a} \oplus \tilde{b} &= (m_1 - \delta_1, m_1, m_1 + \beta_1) \oplus (m_2 - \delta_2, m_2, m_2 + \beta_2) = \\ &= ((m_1 - \delta_1) + (m_2 - \delta_2), m_1 + m_2, (m_1 + \beta_1) + (m_2 + \beta_2)). \end{aligned}$$

### 2.3 Relação de ordem

Visando a comparação entre números *fuzzy*, Okada e Soper [24] propuseram uma relação de ordem para compará-los, conforme descrito na Definição 7.

**Definição 7.** [24] Sejam  $\tilde{a} = (m_1 - \delta_1, m_1, m_1 + \beta_1)$  e  $\tilde{b} = (m_2 - \delta_2, m_2, m_2 + \beta_2)$  dois números *fuzzy* triangulares, então  $\tilde{a} < \tilde{b}$  ( $\tilde{a}$  domina  $\tilde{b}$ ) se, e somente se,  $m_1 - \delta_1 \leq m_2 - \delta_2$ ,  $m_1 \leq m_2$ ,  $m_1 + \beta_1 \leq m_2 + \beta_2$  e  $\tilde{a} \neq \tilde{b}$ .

**Exemplo:** Dados três números triangulares *fuzzy*,  $(a) = \tilde{a} = (m_1 - \delta_1, m_1, m_1 + \beta_1)$ ,  $(b) = \tilde{b} = (m_2 - \delta_2, m_2, m_2 + \beta_2)$  e  $(c) = \tilde{c} = (m_3 - \delta_3, m_3, m_3 + \beta_3)$  - Figura 1.



**Fig. 1.** Dominância de Okada e Soper.

Analisando-os, verifica-se que  $\tilde{a}$  é menor que  $\tilde{c}$ , mas não é possível concluir se  $\tilde{a} < \tilde{b}$ . Porém, aplicando a relação de Def. 7 tem-se  $\tilde{a} < \tilde{b}$  e  $\tilde{a} < \tilde{c}$ . Em algumas casos não é possível detectar se um número domina o outro, assim, no algoritmo proposto os números são mantidos, usando o conceito de etiquetas para armazená-los.

### 3 Algoritmo Proposto

O algoritmo proposto neste trabalho é uma adaptação de Prim [21] e se baseia nos trabalhos de Okada e Soper [24] e Hernandez *et al.* [25], cuja finalidade é tratar de

uma rede com incertezas em seus parâmetros e estrutura. Os pesos das arestas são tratados como números *fuzzy* triangulares, sendo utilizado o conceito de dominância de Okada e Soper [24] para eliminar as arestas de maior peso, enquanto que na estrutura da rede cada aresta possui sua pertinência de fazer parte da rede.

Considerando que o algoritmo proposto tem por objetivo trabalhar com as incertezas até a solução final e apresentar um conjunto solução de árvores geradoras mínimas *fuzzy*, ao contrário do caso clássico em que apenas uma é apresentada, é utilizado o conceito de etiqueta para armazenar as diferentes árvores [24,25].

### 3.1 Passos do algoritmo proposto

O algoritmo é composto de quatro passos, sendo que no primeiro são fornecidos os dados da rede; no segundo a inicialização das variáveis; no terceiro é construído o conjunto solução de árvores geradoras de peso mínimo e no quarto está o critério de parada. No Quadro 1 são apresentados os passos do algoritmo proposto.

**Quadro 1.** Passos do algoritmo proposto

<p><i>Descrição das variáveis:</i>  <math>N</math>: conjunto dos vértices;  <math>A</math>: conjunto das arestas;  <math>\tilde{C}(i, j)</math>: peso da aresta <math>(i, j) \in A</math>;  <math>\mu_{(i,j)}(x)</math>: pertinência da aresta <math>(i,j)</math> pertencer ao grafo <i>fuzzy</i> <math>\tilde{G} = (N, A)</math>;  <math>\mu_{S_k}(x)</math>: pertinência da árvore <math>S_k</math>;  <math>\mu_S(x)</math>: pertinência do conjunto solução;  <math>k</math>: rótulo das árvores geradoras;  <math>T_k</math>: conjunto dos vértices pertencentes à árvore geradora mínima <math>k</math>;  <math>S_k</math>: conjunto das arestas pertencentes à árvore geradora mínima <math>k</math>;  <math>V_k</math>: var. auxiliar que armazena o conjunto dos vértices que não pertencem à árvore geradora mínima <math>k</math>;  <math>n</math>: variável auxiliar que armazena o número de árvores geradoras mínimas;  <math>cont</math>: variável auxiliar que armazena o número de árvores geradoras mínimas do conjunto solução;  <math>aux</math>: variável auxiliar para construção do conjunto solução;  <math>t</math>: variável auxiliar que contabiliza o número de vezes que o Passo 3 se repete a cada iteração;  <math>it</math>: contador de iterações.</p>
<p><b>Algoritmo Proposto</b></p> <p>PASSO 1: Entrada do grafo  Forneça o grafo <i>fuzzy</i> <math>\tilde{G} = (N, A)</math>, com os pesos, <math>\tilde{C}(i, j)</math>, e pertinências, <math>\mu_{(i,j)}(x)</math>, das arestas  <math>it \leftarrow 1</math></p> <p>PASSO 2: Inicialização das variáveis  Escolha o vértice inicial <math>i \in N</math>  <math>n \leftarrow 1</math>  <math>k \leftarrow 1</math>  <math>cont \leftarrow 1</math>  <math>S_k \leftarrow \emptyset</math>  <math>T_k \leftarrow \{i\}</math>  <math>V_k \leftarrow N - \{i\}</math>  <math>t \leftarrow 0</math></p> <p>PASSO 3: Construção do conjunto solução (árvores geradoras de peso mínimo)  <math>t \leftarrow t + 1</math>  Para <math>k \leftarrow 1</math> até <math>n</math> faça  <math>aux \leftarrow \emptyset</math></p>

Continuação do Quadro 1.

```

Para cada aresta  $(i, j) \in A$  tal que  $i \in T_k$  e  $j \in V_k$ 
   $aux \leftarrow aux + \{(i, j)\}$  //verificação das arestas que podem participar da árvore  $T_k$ 
 $(i, j) \leftarrow primeiro(aux)$ 
Enquanto  $(i, j) \neq NIL$  faça
   $(l, m) \leftarrow primeiro(aux)$ 
  Enquanto  $(l, m) \neq NIL$  faça
     $(p, q) \leftarrow sucessor((l, m), aux)$ 
    se  $\tilde{C}(i, j) < \tilde{C}(l, m)$  e  $(i, j) \neq (l, m)$  então // eliminação das arestas de maior peso
       $aux \leftarrow aux - \{(l, m)\}$ 
       $(l, m) \leftarrow (p, q)$ 
       $(i, j) \leftarrow sucessor((i, j), aux)$ 
Para cada aresta  $(i, j) \in aux$  faça // para cada aresta de  $aux$  é construída uma árvore
  Se não existir a árvore  $\{S_k + (i, j)\}$  faça
    Se  $(i, j)$  for a primeira aresta de  $aux$  então
       $r \leftarrow k$ 
    Senão
       $cont \leftarrow cont + 1$ 
       $r \leftarrow cont$ 
       $S_r \leftarrow S_k + \{(i, j)\}$ 
       $T_r \leftarrow T_k + \{j\}$ 
       $V_r \leftarrow V_k - \{j\}$ 
  fim (para)
fim (para-início do Passo 3)

PASSO 4: Cálculo da pertinência de cada árvore e critério de parada
 $Parada \leftarrow verdadeiro$ 
Para  $k \leftarrow 1$  até  $cont$  faça // verificação se cada árvore contém todos os nós
  Se  $|S_k| \neq |N|$ 
     $Parada \leftarrow falso$ 
  Se  $Parada = falso$ 
     $n \leftarrow cont$ 
    Volte ao Passo 3
  Senão
    Para cada  $S_k: k \in 1, 2, \dots, cont$  faça
       $\mu_{S_k}(x) \leftarrow \min \{ \mu_{(i,j)}(x), \forall (i, j) \in S_k \}$ 
       $\mu_S(x) \leftarrow \min \{ \mu_{S_k}(x), k = 1, 2, \dots, cont \}$ 
    Elimine do grafo  $\tilde{G} = (N, A)$  todas as arestas com pert. menor ou igual a  $\mu_S(x)$ 
    Se o novo grafo  $\tilde{G} = (N, A)$  for conexo
       $it \leftarrow it + 1$ 
      volte ao Passo 2
    Senão
      FIM

```

### 3.2 Considerações sobre o algoritmo proposto

Cada vez que o laço principal do Passo 3 se repete um vértice migra do conjunto  $V_k$  para o conjunto  $T_k$  da árvore  $S_k$ . Considerando que  $V_k$  possui inicialmente o vértice  $i$ , o número de vezes que este passo se repete, por iteração, é  $v-1$ , sendo  $v$  o número de vértices do grafo original. Já as iterações se repetem até que o grafo fique desconexo, pois a cada iteração algumas arestas vão sendo eliminados.

Analisando o algoritmo verifica-se que sua aplicação a redes grandes e densas é custosa, haja vista que o conjunto solução é composto de combinações de arestas para gerar o conjunto das árvores geradoras mínimas *fuzzy*. Porém, vale ressaltar que este

conceito foi utilizado em alguns dos principais trabalhos da literatura referentes ao problema do caminho mínimo *fuzzy* [24,26].

## 4 Resultados Computacionais

O algoritmo proposto foi implementado na linguagem Java e testado em duas redes.

### Instância 1

A primeira rede se refere a um grafo de 5 vértices e 10 arestas (Figura 2), sendo que na Tabela 1 estão os pesos e as pertinências das arestas, enquanto que nas Tabelas 2 a 6 estão os resultados obtidos a cada iteração até satisfazer o critério de parada.

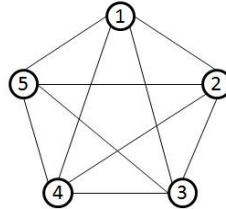


Fig. 2. Rede de Zhou *et al.* [17]

Tabela 1. Dados da rede da Figura 2

Aresta	Origem	Destino	Peso das arestas	Pertinência das arestas
1	1	2	(60, 65, 70)	0,3125
2	2	3	(58, 70, 75)	0,6823
3	3	4	(70, 74, 82)	0,7684
4	4	5	(62, 72, 80)	0,3357
5	1	5	(58, 66, 76)	0,2528
6	1	4	(64,70,74)	0,8132
7	1	3	(58, 62, 70)	0,2987
8	2	5	(65, 70, 80)	0,5255
9	2	4	(55, 60, 65)	0,6349
10	3	5	(62, 68, 72)	0,4966

Tabela 2. Resultados obtidos utilizando todas as arestas do grafo – Iteração 1.

Nº das Arestas da árvore	Peso da árvore	Pertinência da árvore	Pertinência da Iteração
Árvore 1 {1, 5, 7, 9}	(231, 253, 281)	0,2528	
Árvore 2 {1, 7, 9, 10}	(235, 255, 277)	0,2987	0,2528
Árvore 3 {2, 5, 7, 9}	(229, 258, 286)	0,2528	

Tabela 3. Resultados obtidos eliminando as arestas de pertinência  $\leq 0,2528$  – Iteração 2.

Nº das Arestas da árvore	Peso da árvore	Pertinência da árvore	Pertinência da Iteração
Árvore 1 {1, 7, 9, 10}	(235, 255, 277)	0,2987	0,2987
Árvore 2 {2, 7, 9, 10}	(233, 260, 282)	0,2987	

Tabela 4. Resultados obtidos eliminando as arestas de pertinência  $\leq 0,2987$  – Iteração 3.

Nº das Arestas da árvore	Peso da árvore	Pertinência da árvore	Pertinência da Iteração
Árvore 1 {1, 2, 9, 10}	(235, 263, 282)	0,3125	0,3125

**Tabela 5.** Resultados obtidos eliminando as arestas de pertinência  $\leq 0,3125$ – Iteração 4.

	Nº das Arestas da árvore	Peso da árvore	Pertinência da árvore	Pertinência da Iteração
Árvore 1	{2, 6, 9, 10}	(239, 268, 286)	0,4966	0,4966

**Tabela 6.** Resultados obtidos eliminando as arestas de pertinência  $\leq 0,4966$ – Iteração 5.

	Nº das Arestas da árvore	Peso da árvore	Pertinência da árvore	Pertinência da Iteração
Árvore 1	{2, 6, 8, 9}	(242, 272, 294)	0,5255	0,5255

Analisando os resultados verifica-se que à medida que as iterações ocorrem, os pesos das árvores aumentam e as pertinências das mesmas também. Este fato ocorre porque, como consta na Tabela 1, as arestas que possuem os menores pesos (segundo a relação de ordem de Okada e Soper) também contem as menores pertinências, sendo que a cada iteração são eliminadas as arestas com as menores pertinências, logo, a cada iteração o peso das árvores e suas pertinências tendem a aumentar.

Para efeito de comparação, esta instância foi executada usando os algoritmos de Hernandes e Lourenço [15] (pesos das arestas igual ao valor modal), Zhou *et al.* [19] (credibilidade 0,8 e sem pertinências nas arestas) e Prim (pesos das arestas sendo o modal e sem pertinências nas mesmas) obtendo os seguintes resultados (Tabela 7):

**Tabela 7.** Resultados obtidos com algoritmos diversos.

	Nº das Arestas da árvore	Algoritmo (solução)
Árvore 1	{1, 5, 7, 9}	Hernandes e Lourenço (1ª it); Prim
Árvore 2	{1, 7, 9, 10}	Hernandes e Lourenço (2ª it); Zhou <i>et al.</i>
Árvore 3	{1, 2, 9, 10}	Hernandes e Lourenço (3ª it)
Árvore 4	{2, 6, 9, 10}	Hernandes e Lourenço (4ª it)
Árvore 5	{2, 6, 8, 9}	Hernandes e Lourenço (5ª it)

Conforme a Tabela 7, os resultados encontrados por meio dos algoritmos testados pertencem ao conjunto solução do algoritmo proposto (Tabelas 2 a 6), o que auxilia na validação do mesmo.

### Instância 2

A segunda rede testada refere-se a Rede Óptica Italiana de Telecomunicações [27], que possui 21 vértices e 36 arestas (Figura 3).

Nas Tabelas 8 e 9 estão denominados os vértices da Fig. 3 e nas Tabelas 10 a 14 são apresentados os resultados.

**Tabela 8.** Denominação dos vértices da Figura 3

Vértice	Cidade	Vértice	Cidade
1	Bolzano	12	Peragia
2	Milano	13	Roma
3	Verona	14	Pesacara
4	Trieste	15	Napoli
5	Venezia	16	Bari
6	Torino	17	Cagiliani
7	Genova	18	Potenza
8	Bolgano	19	Catanzaro
9	Firenze	20	Palermo
10	Pisa	21	Catamin
11	Ancona		



**Tabela 9.** Pesos e pertinências das arestas da Figura 3

Aresta	Origem	Destino	Peso	Pertinência
1	1	2	(126, 140, 154)	0,59
2	1	3	(99, 110, 121)	0,36
3	1	4	(189, 210, 231)	0,34
4	2	3	(99, 110, 121)	0,43
5	3	5	(81, 90, 99)	0,35
6	4	5	(76, 85, 94)	0,21
7	2	6	(80, 90, 100)	0,61
8	2	7	(85, 95, 105)	0,65
9	3	8	(85, 95, 105)	0,40
10	5	8	85, 95, 105)	0,43
11	6	7	(81, 90, 99)	0,23
12	7	8	(117, 130, 143)	0,65
13	7	9	(135, 150, 165)	0,68
14	7	10	(120, 108, 132)	0,70
15	8	9	(49, 55, 61)	0,15
16	8	11	(180, 200, 220)	0,79
17	9	10	(54, 60, 66)	0,18
18	9	12	(99, 110, 121)	0,52
19	9	13	(162, 180, 198)	0,72
20	10	13	(171, 190, 209)	0,71
21	11	14	(117, 130, 143)	0,69
22	12	14	(153, 170, 187)	0,64
23	12	13	(108, 120, 132)	0,53
24	13	17	(414, 460, 506)	0,91
25	13	15	(162, 180, 198)	0,62
26	14	15	(180, 200, 220)	0,73
27	14	16	(243, 270, 297)	0,74
28	15	16	(189, 210, 231)	0,80
29	15	18	(81, 90, 99)	0,52
30	15	20	(279, 310, 341)	0,78
31	15	21	(315, 350, 385)	0,86
32	16	18	(90, 100, 110)	0,71
33	17	20	(378, 420, 462)	0,87
34	18	19	(180, 200, 220)	0,55
35	19	21	(189, 210, 231)	0,64
36	20	21	(135, 150, 165)	0,35

**Tabela 10.** Resultados obtidos utilizando todas as arestas do grafo – Iteração 1.

	Número das Arestas da árvore	Peso da árvore	Pertinência da árvore	Pertinência da Iteração
Árvore 1	{2, 4, 5, 6, 7, 9, 11, 15, 17, 18, 21, 22, 23, 25, 29, 32, 33, 34, 35, 36}	(2396, 2665, 2934)	0,15	
Árvore 2	{2, 4, 5, 6, 7, 10, 11, 15, 17, 18, 21, 22, 23, 25, 29, 32, 33, 34, 35, 36}	(2396, 2665, 2934)	0,15	0,15

**Tabela 11.** Resultados obtidos eliminando todas as arestas do grafo  $\leq 0,15$  – Iteração 2.

	Número das Arestas da árvore	Peso da árvore	Pertinência da árvore	Pertinência da Iteração
Árvore 1	{2, 4, 5, 6, 7, 9, 11, 14, 17, 18, 21, 22, 23, 25, 29, 32, 33, 34, 35, 36}	(2455, 2730, 3005)	0,18	
Árvore 2	{2, 4, 5, 6, 7, 10, 11, 14, 17, 18, 21, 22, 23, 25, 29, 32, 33, 34, 35, 36}	(2455, 2730, 3005)	0,18	0,18

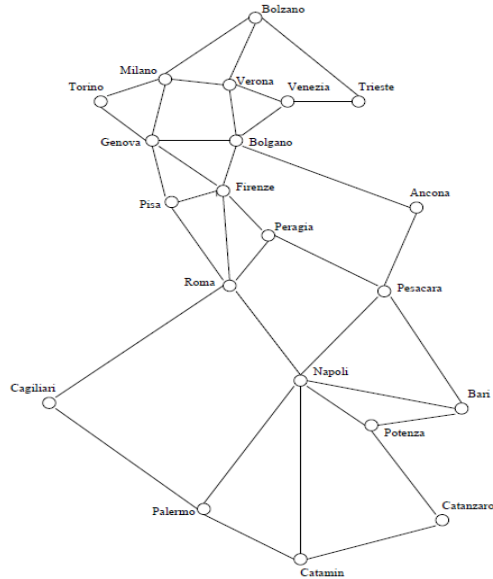


Fig. 3. Rede Óptica Italiana [27].

Tabela 12. Resultados obtidos eliminando todas as arestas do grafo  $\leq 0,18$  – Iteração 3.

	Número das Arestas da árvore	Peso da árvore	Pertinência da árvore	Pertinência da Iteração
Árvore 1	{2, 4, 5, 6, 7, 9, 11, 13, 14, 18, 21, 22, 23, 25, 29, 32, 33, 34, 35, 36}	(2536, 2820, 3104)	0,21	0,21
Árvore 2	{2, 4, 5, 6, 7, 10, 11, 13, 14, 18, 21, 22, 23, 25, 29, 32, 33, 34, 35, 36}	(2536, 2820, 3104)	0,21	

Tabela 13. Resultados obtidos eliminando todas as arestas do grafo  $\leq 0,21$  – Iteração 4.

	Número das Arestas da árvore	Peso da árvore	Pertinência da árvore	Pertinência da Iteração
Árvore 1	{2, 3, 4, 5, 7, 9, 11, 13, 14, 18, 21, 22, 23, 25, 29, 32, 33, 34, 35, 36}	(2649, 2945, 3241)	0,23	0,23
Árvore 2	{2, 3, 4, 5, 7, 10, 11, 13, 14, 18, 21, 22, 23, 25, 29, 32, 33, 34, 35, 36}	(2649, 2945, 3241)	0,23	

Tabela 14. Resultados obtidos eliminando todas as arestas do grafo  $\leq 0,23$  – Iteração 5.

	Número das Arestas da árvore	Peso da árvore	Pertinência da árvore	Pertinência da Iteração
Árvore 1	{2, 3, 4, 5, 7, 8, 9, 13, 14, 18, 21, 22, 23, 25, 29, 32, 33, 34, 35, 36}	(2653, 2950, 3247)	0,34	0,34
Árvore 2	{2, 3, 4, 5, 7, 8, 10, 13, 14, 18, 21, 22, 23, 25, 29, 32, 33, 34, 35, 36}	(2653, 2950, 3247)	0,34	

Assim como na Instância 1, verifica-se que à medida em que as iterações ocorrem o peso das árvores aumentam, pois as arestas de menor pertinência têm também os menores pesos.

Para efeito de comparação, nesta instância também foram executados os mesmos algoritmos da Instância 1, com os mesmos parâmetros, sendo obtidos os resultados da Tabela 15.

**Tabela 15.** Resultados obtidos com algoritmos diversos.

	<b>Arestas da árvore</b>	<b>Algoritmo (solução)</b>
Árvore 1	{2, 4, 5, 6, 7, 9, 11, 15, 17, 18, 21, 22, 23, 25, 29, 32, 33, 34, 35, 36}	Hernandes e Lourenço (1ª it); Prim; Zhou <i>et al.</i>
Árvore 2	{2, 4, 5, 6, 7, 9, 11, 14, 17, 18, 21, 22, 23, 25, 29, 32, 33, 34, 35, 36}	Hernandes e Lourenço (2ª it)
Árvore 3	{2, 4, 5, 6, 7, 9, 11, 13, 14, 18, 21, 22, 23, 25, 29, 32, 33, 34, 35, 36}	Hernandes e Lourenço (3ª it)
Árvore 4	{2, 3, 4, 5, 7, 9, 11, 13, 14, 18, 21, 22, 23, 25, 29, 32, 33, 34, 35, 36}	Hernandes e Lourenço (4ª it)
Árvore 5	{2, 3, 4, 5, 7, 8, 9, 13, 14, 18, 21, 22, 23, 25, 29, 32, 33, 34, 35, 36}	Hernandes e Lourenço (5ª it)

Analisando a Tabela 15 verifica-se que os resultados encontrados por meio dos algoritmos de Hernandez e Lourenço [15], Zhou *et al.* [19] e Prim pertencem ao conjunto solução do algoritmo proposto.

## 5 Conclusões e Trabalhos Futuros

O problema da árvore geradora mínima possui aplicações em diversas áreas da Engenharia e da Informática, sendo que em muitos casos estes problemas possuem incertezas nos pesos das arestas e até mesmo na estrutura da rede. Logo, o objetivo deste trabalho foi abordar o problema da árvore geradora mínima *fuzzy*, propondo um algoritmo que trabalhe com as incertezas nos parâmetros e na estrutura até a obtenção da solução final.

Considerando a dificuldade em comparar números *fuzzy*, o algoritmo apresentado é uma adaptação do algoritmo clássico de Prim, utilizando a relação de dominância de Okada e Soper para eliminar as arestas com os maiores pesos, bem como o conceito de rótulo para armazenar as árvores do conjunto solução. Trata-se de algoritmo exato que trabalha com as incertezas nos pesos e na estrutura até a obtenção da solução final, sendo uma abordagem inédita da literatura. Ressalta-se que o desempenho do mesmo foi compatível aos da literatura, como pôde ser visto por meio dos resultados.

Com relação ao problema abordado, PAGM, ressalta-se que se trata de um problema de minimização, porém, devido a relação de ordem utilizada e por se tratar de dados incertos, nem sempre a solução final encontrada foi única, pois em algumas comparações não foi possível encontrar uma única árvore.

Como trabalhos futuros pretende-se abordar outras relações de ordem, bem como propor um índice de ordenação que leve em consideração os pesos das árvores e suas pertinências, como por exemplo, uma média entre os dois.

## Referências

1. Chunde, Y.: On the Optimization Problem of Spanning Tree in Fuzzy Network. The Journal of China Universities of Posts and Telecommunications 3 (2), 22-25 (1996).
2. Chang, P.T., Lee, E.S.: Fuzzy Decision Networks and Deconvolution. Computers and Mathematics with Applications 37 (11-12), 53-63 (1999).

3. Raidl, G.R., Julstrom, B.A.: Edge Sets: An Effective Evolutionary Coding of Spanning Trees. *IEEE Transactions on Evolutionary Computation* 7 (3), 225-239 (2003).
4. Ahuja, R., Magnati, T., Orlin, J.: *Network Flows: theory, algorithms and applications*. Prentice-Hall, New York (1993).
5. Takahashi, M.T., Yamakami, A.: Um Estudo sobre Árvore Geradora Mínima com Parâmetro Fuzzy. In: XXXV Simpósio Brasileiro de Pesquisa Operacional, 2217-2227 (2003).
6. Takahashi, M.T., Yamakami, A.: Um Estudo sobre o Problema da Árvore Geradora Mínima com Estrutura do Grafo Fuzzy. In: XXXVI Simpósio Brasileiro de Pesquisa Operacional, 2278-2285 (2004).
7. Zadeh, L.: Fuzzy Sets. *Information and Control* 8, 338-353 (1965).
8. Zadeh, L.: Fuzzy Algorithms. *Information and Control* 12 (1968).
9. Zadeh, L.: Fuzzy Sets as a Theory of Possibility. *Journal of Fuzzy Sets and Systems* 1 (1978).
10. Delgado, M., Verdegay, J.L., Vila, M.A.: On Fuzzy Tree Definition. *European Journal of Operational Research* 22, 243-249 (1985).
11. Janiak, A., Kasperski, A.: The Minimum Spanning Tree Problem with Fuzzy Costs. *Fuzzy Optimization and Decision Making* 7, 105-118 (2008).
12. Mohanty, S.P., Biswal, S., Pradhan, G.: Minimum Spanning Tree in Fuzzy Weighted Rough Graph. *International Journal of Engineering Research and Development* 1(10), 23-28 (2012).
13. Nayeem, S.M.A., Pal, M.: Diameter Constrained Fuzzy Minimum Spanning Tree Problem. *International Journal of Computational Intelligence Systems* 6(6), 1040-1051 (2013).
14. Dhanalakshmi, S., Parvathi, N.: Minimum Spanning Tree of a Rough Graph Having Weights as Trapezoidal Fuzzy Numbers. *Global Journal of Pure and Applied Mathematics* 11(1), 143-147 (2015).
15. Hernandez, F., Lourenço, M. H. R. S.: An Algorithm for the Minimum Spanning Tree Problem with Uncertain Structures. *IEEE Latin America Transactions* 13(12), 3885-3889 (2015).
16. Gao, X.: Tree Index of Uncertain Graphs. *Soft Computing* 20, 1449-1458 (2016).
17. Theresa, M. M. J., Raj, V. J.: A Maximum Spanning Tree-Based Dynamic Fuzzy Supervised Neural Network Architecture for Classification of Murder Cases. *Soft Computing* 20, 2353-2365 (2016).
18. Zhang, X., Zhang, X.: A Binary Artificial Bee Colony Algorithm for Constructing Spanning Trees in Vehicular Ad Hoc Networks. *Ad Hoc Networks*, 1-7 (2016).
19. Zhou, J. et al.: Fuzzy  $\alpha$ -Minimum Spanning Tree Problem: Definition and Solutions. *International Journal of General Systems* 45 (3), 311-335 (2016).
20. Yao, J.S., Lin, F.T.: Fuzzy Shortest Path Network Problems with Uncertain Edge Weight. *Journal of Information Science and Engineering* 19, 329-351 (2003).
21. Gondran, M., Minoux, M.: *Graphs and Algorithms*. John Wiley & Sons, New York (1984).
22. Pedrycz, W., Gomide, F.: *Fuzzy Systems Engineering: Toward Human-Centric Computing*. IEEE/Wiley Interscience (2007).
23. Pedrycz, W., Gomide, F.: *An Introduction to Fuzzy Sets: Analysis and Design*. MIT Press, London (1998).
24. Okada, S., Soper, T.: A Shortest Path Problem on a Network with Fuzzy Arc Lengths. *Fuzzy Sets and Systems* 109, 129-140 (2000).
25. Hernandez, F., Lamata, M.T., Verdegay, J.L., Yamakami, A.: The Shortest Problem on Networks with Fuzzy Parameters. *Fuzzy Sets and Systems* 158, 1561-1570 (2007).
26. Okada, S.: Fuzzy Shortest Path Problems Incorporating Interactivity Among Paths. *Fuzzy Sets and Systems* 142 (3), 335-357 (2004).
27. Ali, M., Ramamurthy, B., Deogun, J.: Routing and Wavelength Assignment with Power Considerations in Optical Networks. *Computer Networks* 32, 539-555 (2000).