

O Uso da Dominância no Problema da Árvore Geradora Mínima com Parâmetros *Fuzzy*

The Dominance Concept Applied in the Minimum Spanning Tree Problem with Fuzzy Parameters

Fabio Hernandes

Departamento de Ciência da Computação
Universidade Estadual do Centro-Oeste – UNICENTRO, Guarapuava, PR
hernandes.fabio@gmail.com

Lucas Fernando Frighetto

Departamento de Ciência da Computação
Universidade Estadual do Centro-Oeste – UNICENTRO, Guarapuava, PR
lucas.frighetto@gmail.com

Mauro Henrique Mulati

Departamento de Ciência da Computação
Universidade Estadual do Centro-Oeste – UNICENTRO, Guarapuava, PR
mhmulati@gmail.com

Resumo: O problema da árvore geradora mínima com parâmetros *fuzzy* é um dos principais problemas da programação matemática *fuzzy*, visto que possui aplicações nas mais diversas áreas, como, por exemplo, redes de distribuição de energia, armazenamento de informações e transportes. Neste trabalho é proposto um algoritmo exato para o problema da árvore geradora mínima com estrutura *crisp* e parâmetros *fuzzy*. Trata-se de uma adaptação do algoritmo clássico de Prim, utilizando a relação de dominância de alguns trabalhos da literatura que tratam do problema de caminho mínimo *fuzzy*. As incertezas nos pesos são abordadas por meio da teoria dos conjuntos *fuzzy* e as árvores geradoras do conjunto solução são armazenadas usando o conceito de rótulos. O algoritmo proposto foi testado em três diferentes redes, compatíveis às instâncias dos principais trabalhos da literatura.

Recebido em 17/08/2015 - Aceito em 11/04/2016.

RECEN 18(1) p. 31-54 jan/jun 2016 DOI: 10.5935/RECEN.2016.01.02

Palavras-chave: algoritmo de Prim; programação matemática; teoria de grafos; teoria dos conjuntos *fuzzy*.

Abstract: The minimum spanning tree problem with fuzzy parameters is one of the main problems in fuzzy mathematical programming, as it has applications in several areas, such as, power distribution networks, information storage and transportation. This work proposed an exact algorithm to the problem of minimum spanning tree problem with crisp structure and fuzzy parameters. It is an adaptation of the classic algorithm Prim, using the dominance relationship based on some literature review dealing with the fuzzy shortest path problem. The uncertainties in the weights are addressed using the Fuzzy Set Theory, while the spanning trees in the solution set are stored using the concept of labels. The proposed algorithm was tested on three different networks, compatible with the instances found in the main literature reviews in the area.

Keywords: fuzzy sets theory; graphs theory; mathematical programming; Prim's algorithm.

1 Introdução

Incertezas, imprecisões e ambiguidades são fatores que não podem ser ignorados na resolução de alguns problemas reais visto que, em muitos casos, os parâmetros (custos, capacidades, demandas, dentre outros) podem não ser naturalmente precisos. Com isso, nas décadas de 1960 e 1970, Zadeh [1–3] introduziu a teoria dos conjuntos *fuzzy*, cuja finalidade foi dar um tratamento matemático a termos linguísticos subjetivos como “aproximadamente” e “em torno de” dentre outros.

A teoria dos grafos é uma das áreas da programação matemática que tem utilizado frequentemente a teoria dos conjuntos *fuzzy* em alguns de seus problemas, dentre os quais se destacam: caminho mínimo [4–11], fluxo máximo [12–15], fluxo de custo mínimo [12, 16, 26], árvore geradora mínima [17–21] e coloração de grafos [22–25]. Dentre estes problemas, o da árvore geradora mínima é um dos principais, haja vista

suas aplicações na Engenharia e na Informática, tais como: instalações e projeções de redes de telecomunicações (redes de computadores, redes de telefonia e redes de TV a cabo) e elétricas, problemas de transporte (rodovias e ferrovias), dentre outros [26].

Na literatura são apresentados diversos algoritmos que resolvem o problema clássico da árvore geradora mínima sendo, principalmente, os algoritmos de Kruskal, Prim, Borüvk e Sollin [27]. Porém, ao considerar incertezas nos parâmetros ou na estrutura da rede, a resolução desse problema se torna mais trabalhosa, pois a comparação entre os dados dos parâmetros não é trivial como no caso clássico, reportando assim à teoria dos conjuntos *fuzzy*. O número de trabalhos na literatura que trata do tema é reduzido, devendo-se principalmente a Delgado et al [28], Chunde [29], Takahashi e Yamakami [18], Takahashi e Yamakami [17], Gao e Lu [30], Janiak e Kasperski [31], Mohanty et al [19] e Nayeem e Pal [21].

No trabalho de Delgado et al [28] foram introduzidos os conceitos do problema da árvore geradora mínima em redes com parâmetros e/ou estruturas *fuzzy* e utilizado o conceito de α -corte para obtenção de cada árvore geradora mínima do conjunto *fuzzy* de soluções. Chunde [29] propôs, por meio de uma heurística, um algoritmo em que é realizada apenas uma iteração do algoritmo de Delgado et al [28], que utiliza o α -corte mais conveniente, aplicado a redes com incertezas em sua estrutura. Takahashi e Yamakami [18] propuseram um algoritmo genético para o problema com parâmetros *fuzzy*, usando a teoria de possibilidade de Zadeh [3], enquanto que no segundo trabalho [17] foi abordado o problema em uma rede com estrutura *fuzzy*, adaptando os trabalhos de Delgado et al [28] e Chunde [29]. Gao e Lu [30] abordaram o problema da árvore geradora mínima quadrática *fuzzy* com parâmetros incertos e propuseram um algoritmo genético para solucioná-lo. Janiak e Kasperski [31] aplicaram a teoria da possibilidade de Zadeh [3] nos custos das arestas e utilizaram um algoritmo clássico da literatura para encontrar a árvore geradora mínima. Mohanty et al [19] abordaram o problema com parâmetros *fuzzy* por meio do algoritmo clássico de Borüvka, juntamente com o índice de ordenação de Yao e Lin [32]. Em um dos trabalhos mais recentes, Nayeem e Pal [21] “defuzzificaram”, o problema, transformando-o em um problema clássico, e o resolveram via algoritmos genéticos.

Analisando os trabalhos descritos anteriormente, verifica-se que estes apresentam

algumas particularidades, tais como: solucionam o problema via algoritmos genéticos [17, 18, 21, 29, 30] ou aplicam índices de “defuzzificação”, resolvendo o problema por meio de métodos clássicos [19, 31], o que pode ocasionar perda de informações. Além do mais, nos trabalhos descritos anteriormente seus algoritmos são aplicados em redes de pequeno porte (máximo de 21 vértices e 36 arestas, também utilizada neste trabalho), pouco densas e com várias arestas dominadas (descartadas), o que faz com que o uso de métodos heurísticos não seja totalmente justificável. Com isso, o objetivo deste trabalho é apresentar um algoritmo exato, adaptado do algoritmo de Prim, para resolver o problema da árvore geradora mínima em redes com parâmetros *fuzzy*, aplicável principalmente a redes de pequeno porte (similares às utilizadas nos trabalhos da literatura) ou pouco densas e com arestas dominadas (arestas caras, de peso grande).

O algoritmo proposto trabalha com as incertezas até a obtenção da solução final, aplicando, assim, a relação de dominância de Okada e Soper [4] para eliminar as arestas de maior peso e o conceito de rótulos para armazenar o conjunto solução, pois o mesmo pode apresentar mais de uma árvore geradora mínima. Trata-se de um algoritmo baseado nos trabalhos de Okada e Soper [4], Okada [6] e Hernandez et al [8].

Este trabalho está organizado da seguinte forma: na seção 2 são apresentados os conceitos e definições utilizados no trabalho; a seção 3 aborda o algoritmo proposto; na seção 4 estão os resultados computacionais, enquanto que na seção 5 são destacadas as conclusões e trabalhos futuros.

2 Conceitos e definições

Nesta seção são abordados alguns conceitos e definições utilizados no trabalho. Inicialmente é introduzido o problema da árvore geradora de peso mínimo, seguido de alguns conceitos da teoria dos conjuntos *fuzzy* e do problema da árvore geradora mínima *fuzzy*.

2.1 O problema da árvore geradora mínima

Definição 1. Um grafo não orientado $G = (N, A)$, tal que N é o conjunto dos vértices e A o conjunto das arestas, é uma *árvore* se, e somente se, G for conexo e acíclico [33].

Seja um grafo $G = (N, A)$, tal que para cada aresta $(i, j) \in A$ associa-se um número C_{ij} , chamado *peso da aresta* (i, j) . 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.

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 .

Teorema 1. Se todas as arestas do grafo $G = (N, A)$ possuírem pesos distintos, a árvore de peso mínimo é única [33].

2.2 Teoria dos conjuntos *fuzzy*

Considerando que os pesos e as arestas das redes tratadas no algoritmo proposto são *fuzzy*, nesta subseção são apresentados alguns conceitos de números *fuzzy*, bem como a relação de ordem utilizada para eliminar as arestas consideradas caras (pesos maiores).

2.2.1 Números *fuzzy*

As definições apresentadas são encontradas em Dubois e Prade [34] e Pedrycz e Gomide [35].

Considerando que os pesos das arestas utilizados neste trabalho são abordados como números *fuzzy* triangulares, as definições apresentadas tratam deste número.

Definição 2. Um *número fuzzy triangular*, denotado por $\tilde{a} = (m, \alpha, \beta)$, possui sua função de pertinência, $\mu_{\tilde{a}}(x)$, definida da seguinte forma (Figura 1):

$$\mu_{\tilde{a}}(x) = \begin{cases} 0, & \text{se } x \leq m - \alpha \\ \frac{x - (m - \alpha)}{\alpha}, & \text{se } m - \alpha < x < m \\ 1, & \text{se } x = m \\ \frac{(m + \beta) - x}{\beta}, & \text{se } m < x < m + \beta \\ 0, & \text{se } x \geq m + \beta \end{cases}$$

tal que: m é o valor modal, α o espalhamento à esquerda e β o espalhamento à direita.

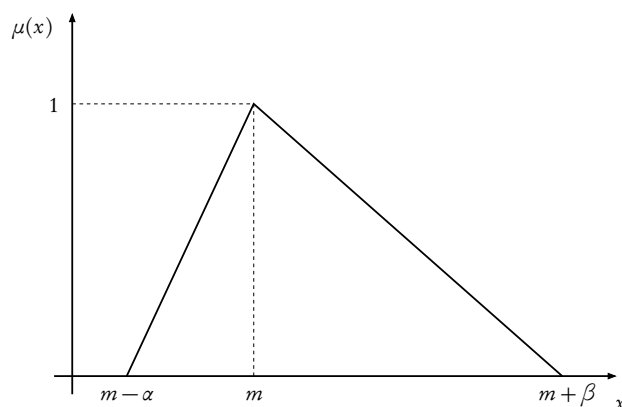


Figura 1. Exemplo de uma função de pertinência triangular

Definição 3. Valor modal é o valor $x \in [m - \alpha, m + \beta]$ para o qual a função de pertinência tem valor máximo.

Definição 4. Sejam \tilde{a} e \tilde{b} dois números *fuzzy* triangulares, sendo $\tilde{a} = (m_1, \alpha_1, \beta_1)$ e $\tilde{b} = (m_2, \alpha_2, \beta_2)$, a soma *fuzzy* entre \tilde{a} e \tilde{b} é denotada por:

$$\begin{aligned} \tilde{a} \oplus \tilde{b} &= (m_1, \alpha_1, \beta_1) \oplus (m_2, \alpha_2, \beta_2) = \\ &= (m_1 + m_2, \alpha_1 + \alpha_2, \beta_1 + \beta_2) \end{aligned}$$

2.2.2 Relação de ordem

Okada e Soper [4] abordaram, em seu trabalho, o problema do caminho mínimo *fuzzy* e, para eliminar os caminhos considerados "caros", propuseram uma relação de ordem baseada nas operações "*fuzzy max*" e "*fuzzy min*", definidas pelo princípio da extensão, como segue:

$$\mu_{\max\{\tilde{a}, \tilde{b}\}}(r) = \sup_{r=\max\{s,t\}} \{\mu_{\tilde{a}}(s), \mu_{\tilde{b}}(t)\}$$

tal que: $\mu_{\max\{\tilde{a}, \tilde{b}\}}$ é uma função de pertinência $\{\tilde{a}, \tilde{b}\}$ e s, t são os nós origem e destino, respectivamente.

O máximo de n números *fuzzy* $\tilde{a}_1, \tilde{a}_2, \dots, \tilde{a}_n$, denotado por $\widetilde{\max}\{\tilde{a}_1, \tilde{a}_2, \dots, \tilde{a}_n\}$, é também um número *fuzzy*, mas $\widetilde{\max}\{\tilde{a}_1, \tilde{a}_2, \dots, \tilde{a}_n\}$ não necessariamente é um dos \tilde{a}'_i s. Logo, a relação de ordem de Okada e Soper [4] é:

Definição 5. Dados dois números *fuzzy* triangulares, $\tilde{a} = (m_1, \alpha_1, \beta_1)$ e $\tilde{b} = (m_2, \alpha_2, \beta_2)$, $\tilde{a} < \tilde{b}$ (\tilde{a} domina \tilde{b}) se, e somente se, $m_1 \leq m_2$, $(m_1 - \alpha_1) \leq (m_2 - \alpha_2)$, $(m_1 + \beta_1) \leq (m_2 + \beta_2)$ e $\tilde{a} \neq \tilde{b}$.

Exemplo. Dados três números *fuzzy* triangulares $\tilde{a} = (m_1, \alpha_1, \beta_1)$, $\tilde{b} = (m_2, \alpha_2, \beta_2)$ e $\tilde{c} = (m_3, \alpha_3, \beta_3)$, conforme figura 1.

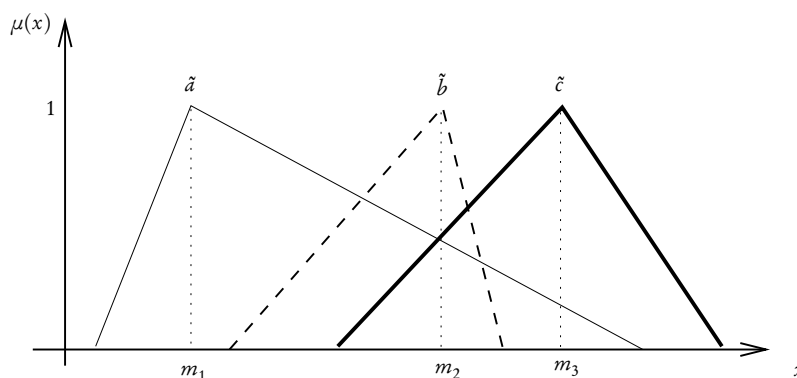


Figura 2. Exemplo de dominância

Aplicando a relação de dominância de Okada e Soper (Definição 5) verifica-se que

$\tilde{a} \not\prec \tilde{b}$, $\tilde{a} \prec \tilde{c}$, $\tilde{b} \not\prec \tilde{a}$, $\tilde{b} \prec \tilde{c}$, $\tilde{c} \not\prec \tilde{a}$ e $\tilde{c} \not\prec \tilde{b}$. Logo, \tilde{c} é eliminado, pois foi dominado por, pelo menos, um dos outros números, isto significa que possui peso maior que, no mínimo, um dos outros.

2.3 O problema da árvore geradora mínima *fuzzy*

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:

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

O *problema da árvore mínima fuzzy* consiste em procurar uma árvore T^* de \tilde{G} , 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} .

3 Algoritmo proposto

O algoritmo proposto é uma adaptação do algoritmo clássico de Prim [33] e se baseia nos trabalhos de Okada e Soper [4] e Hernandez et al [8], cuja finalidade é tratar de uma rede com incertezas em seus parâmetros. Os pesos das arestas são tratados como números *fuzzy* triangulares e foi utilizado o conceito de dominância de Okada e Soper [4] para eliminar as arestas de maior peso.

Considerando que o algoritmo proposto tem por objetivo 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 rótulo para armazenar as diferentes árvores do conjunto solução, similar aos trabalhos de caminho mínimo *fuzzy* de Okada e Soper [4] e Hernandez et al [8].

O algoritmo proposto é composto de três passos, a saber: no Passo 0 são inicializadas as variáveis; no Passo 1 são construídas as árvores geradoras de peso mínimo,

sendo que inicialmente são verificadas todas as possíveis arestas que podem fazer parte da árvore (constituição da variável aux) e, em seguida, são eliminadas as de maior peso (aplicação da dominância). Com as arestas restantes são construídas as árvores geradoras (uma árvore para cada aresta de aux). O Passo 2 é o critério de parada, sendo verificado se todas as possíveis árvores são realmente árvores do grafo $\tilde{G} = (N, A)$, caso contrário, retorna-se ao Passo 1.

3.1 Passos do algoritmo proposto

Descrição das variáveis:

N : conjunto dos vértices;

A : conjunto das arestas;

$\tilde{C}(i, j)$: peso da aresta $(i, j) \in A$;

k : rótulo das árvores geradoras;

T^k : conjunto dos vértices pertencentes à árvore geradora mínima k ;

S^k : conjunto das arestas pertencentes à árvore geradora mínima k ;

V^k : variável auxiliar que armazena o conjunto dos vértices que não pertencem à árvore geradora mínima k ;

n : variável auxiliar que armazena o número de árvores geradoras mínimas;

$cont$: variável auxiliar que armazena o número de árvores geradoras mínimas do conjunto solução;

aux : variável auxiliar para construção do conjunto solução; e

it : contador de iterações.

Algoritmo Proposto

PASSO 0: Inicialização das variáveis

Forneça o grafo $\tilde{G} = (N, A)$, com seus respectivos pesos $\tilde{C}(i, j)$ das arestas

Escolha o vértice inicial $i \in N$

$k \leftarrow 1$

$n \leftarrow 1$

$cont \leftarrow 1$
 $S^k \leftarrow \emptyset$
 $T^k \leftarrow \{i\}$
 $V^k \leftarrow N - \{i\}$
 $it \leftarrow 0$

PASSO 1: Construção do conjunto solução (árvores geradoras de peso mínimo)

$it \leftarrow it + 1$

Para $k \leftarrow 1$ até n faça

$aux \leftarrow \emptyset$

Para cada aresta $(i, j) \in A$ tal que $i \in T^k$ e $j \in V^k$

$aux \leftarrow aux + \{(i, j)\}$

$(i, j) \leftarrow \text{primeiro}(aux)$

Enquanto $(i, j) \neq NIL$ faça

$(l, m) \leftarrow \text{primeiro}(aux)$

Enquanto $(l, m) \neq NIL$ faça

$(p, q) \leftarrow \text{sucessor}((l, m), aux)$

se $\tilde{C}(i, j) < \tilde{C}(l, m)$ e $(i, j) \neq (l, m)$ então

$aux \leftarrow aux - \{(l, m)\}$

$(l, m) \leftarrow (p, q)$

$(i, j) \leftarrow \text{sucessor}((i, j), aux)$

Para cada aresta $(i, j) \in aux$ faça

Se não existir a árvore $\{S^k + (i, j)\}$ faça

Se (i, j) for a primeira aresta de aux então

$r \leftarrow k$

Senão

$cont \leftarrow cont + 1$

$r \leftarrow cont$

$S^r \leftarrow S^k + \{(i, j)\}$

$T^r \leftarrow T^k + \{j\}$

$V^r \leftarrow V^k - \{j\}$

fm(para- início do Passo 1)

PASSO 2: Critério de Parada

Parada \leftarrow verdadeiro

Para $k \leftarrow 1$ até $cont$ faça

 Se $|S^k| \neq (|N| - 1)$

 Parada \leftarrow falso

Se Parada = falso

$n \leftarrow cont$

 Volte ao Passo 1

Senão

 Retorne $\{S^k : k \in 1, 2, \dots, cont\}$

FIM

3.2 Considerações sobre o algoritmo proposto

A cada iteração do algoritmo proposto um vértice migra do conjunto V^k para o conjunto T^k da árvore S^k . Logo, considerando que V^k possui inicialmente o vértice i , o número de iterações do Passo 1 do algoritmo é $v-1$, sendo v o número de vértices do grafo original.

Analisando o algoritmo proposto verifica-se que sua aplicação a grandes redes, densas e com pouca dominância pode não ser viável, haja vista que o conjunto solução é composto de combinações de arestas para gerar o conjunto das árvores geradoras mínimas *fuzzy*. Porém, vale destacar que os principais trabalhos da literatura abordaram o problema em redes pouco densas e com, pelo menos, 60% de arestas dominadas, o que justifica a aplicação deste algoritmo. Também, ressalta-se que o conceito de dominância foi utilizado em alguns trabalhos da literatura referentes ao problema do caminho mínimo com incertezas, trabalhos estes que apresentaram algoritmos exatos [4,6], semelhantes ao proposto, e que foram publicados nos principais periódicos da área, além de serem bem citados na literatura.

4 Resultados computacionais

O algoritmo proposto foi implementado em Java e executado em três redes distintas não orientadas, sendo as duas primeiras com pesos fictícios.

4.1 Instância 1

Nesta instância foi considerada uma rede pequena com 5 vértices e 7 arestas (Figura 3), a fim de detalhar as árvores obtidas passo a passo na execução do algoritmo.

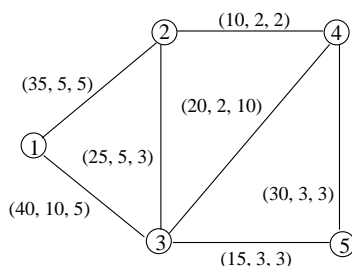


Figura 3. Rede 1

A seguir (Figuras 4-7) são apresentadas as arestas selecionadas (em negrito) a cada iteração do Passo 1 aplicado na rede da figura 3, considerando o vértice 1 como inicial.

Na primeira iteração poderiam ser selecionadas as arestas (1,2) e (1,3), porém a escolhida foi a (1,2) que dominou a outra (Figura 4).

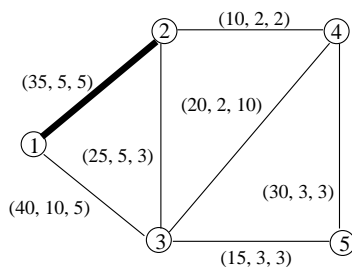


Figura 4. Aresta selecionada após a primeira iteração – Árvore S^1

Na segunda iteração as possíveis arestas eram (1,3), (2,3) e (2,4). A selecionada foi a aresta (2,4), que dominou as demais (Figura 5).

Na terceira iteração do Passo 1 as arestas possíveis eram (1,3), (2,3), (4,3) e (4,5), sendo as arestas (2,3) e (4,3) não dominadas. Com isso, a variável S^1 deu origem a mais

uma variável (S^2), originando assim duas possíveis árvores geradoras de peso mínimo, S^1 e S^2 , conforme figura 6.

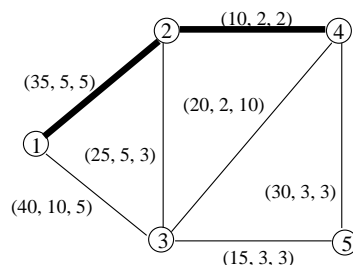
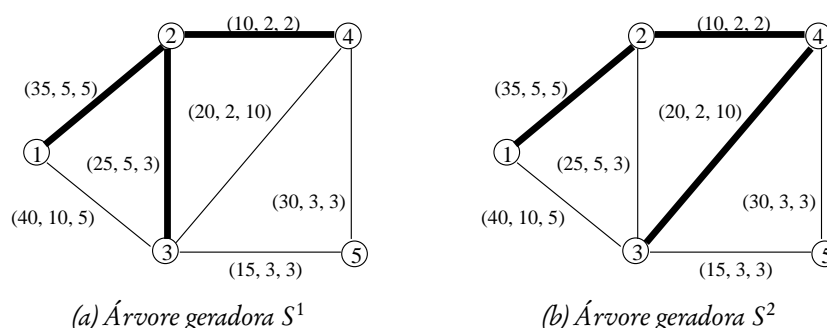


Figura 5. Arestas selecionadas após a segunda iteração – Árvore S^1



(a) Árvore geradora S^1

(b) Árvore geradora S^2

Figura 6. Árvores geradoras mínimas fuzzy obtidas na terceira iteração – Instância 1

Na quarta e última iteração as possíveis arestas eram (3,5) e (4,5), sendo a (3,5) selecionada, pois dominou a outra, e incluída nas variáveis S^1 e S^2 . As duas árvores geradoras - S^1 e S^2 (Figura 7) - do conjunto solução apresentaram pesos finais de (85, 15, 13) e (80, 12, 20), respectivamente.

Ao analisar o grafo da Rede 1, verifica-se que sua densidade, D , é de 70% ($D = (2*A)/(v*(v-1))$, tal que A é o número de aresta e v o número de vértices do grafo), o que o caracteriza como denso, porém por se tratar de uma rede pequena o algoritmo encontrou todas as árvores geradoras mínimas fuzzy.

Observando as árvores geradoras do conjunto solução verifica-se que a única diferença entre elas está nas arestas (2,3) e (4,3), sendo que a primeira está na árvore S^1 e a segunda na árvore S^2 . Com relação aos pesos das mesmas, conclui-se: (i) a árvore S^2 possui o menor espalhamento à esquerda (limitante inferior) e o menor modal,

podendo ser escolhida pelo usuário que opte por um destes índices; (ii) a árvore S^1 possui o menor espalhamento à direita (limitante superior), podendo ser escolhida pelo usuário que estiver interessado neste índice.

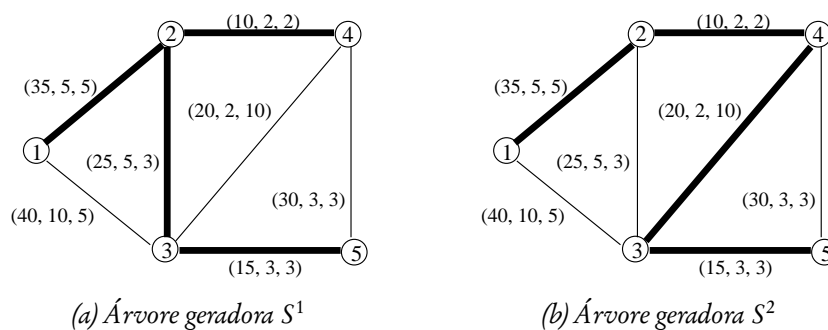


Figura 7. Conjunto final das árvores geradoras mínimas fuzzy – Instância 1

A figura 8 apresenta, resumidamente, como foram obtidas as duas árvores geradoras mínimas ao longo das iterações desta instância.

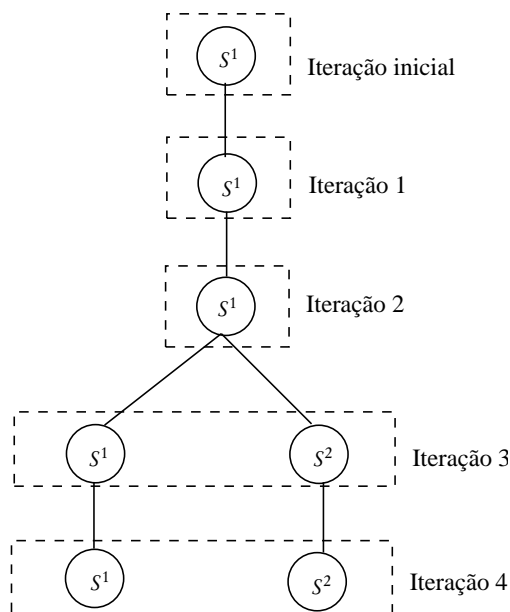


Figura 8. Obtenção das árvores geradoras ao longo das iterações – Instância 1

4.2 Instância 2

Nesta instância foi considerada uma rede com 6 vértices e 9 arestas de Hernandes et al [8], com densidade de 60%, sendo a aresta (4,5) com peso negativo (Figura 9).

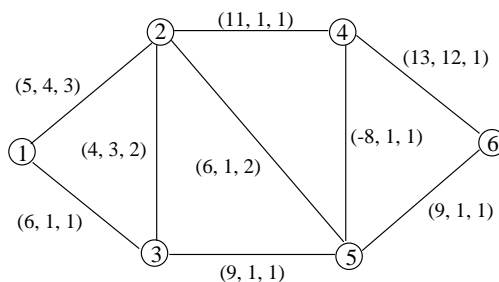


Figura 9. Rede 2 – peso negativo

Ao executar o algoritmo implementado foi escolhido o vértice 1 como origem, sendo que o mesmo executou 5 iterações, obtendo como solução final um conjunto com quatro árvores geradoras (Figura 10), S^1, S^2, S^3 e S^4 , com os pesos (20, 21, 9), (21, 18, 7), (16, 10, 9) e (17, 7, 7), respectivamente.

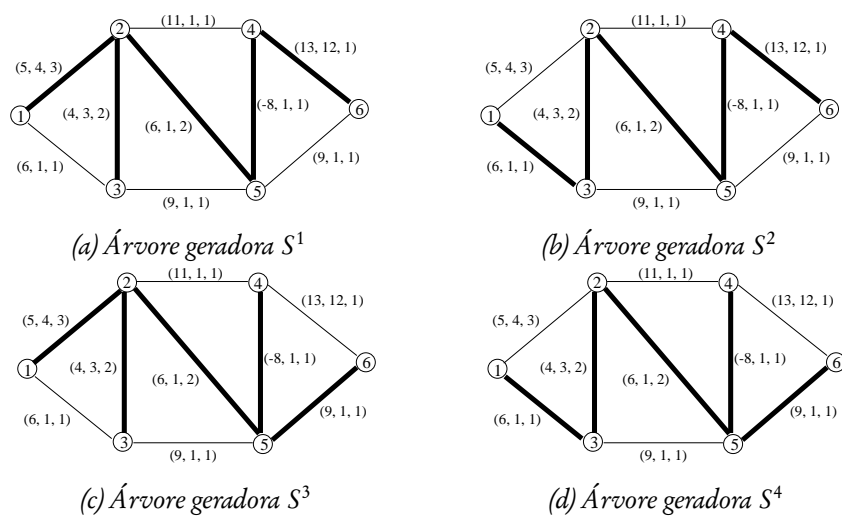


Figura 10. Árvores geradoras mínimas fuzzy obtidas na Instância 2

No conjunto solução encontrado verifica-se que as arestas (2,3), (2,5) e (4,5) são comuns entre elas, isso se justifica porque a aresta (2,3) é a de menor peso que incide

no nó 3, a (2,5) é a de menor peso que une o conjunto formado pelos nós 1, 2 e 3 ao conjunto formado pelos nós 4, 5 e 6. Já a aresta (4,5) está presente em todas as árvores porque é a que possui o menor peso de todas (peso negativo).

Ao analisar os pesos das árvores do conjunto solução conclui-se: (i) caso o usuário esteja interessado no espalhamento à esquerda (limitante inferior), escolherá a árvore S^1 ; (ii) caso esteja interessado no menor modal, a árvore escolhida será a S^3 ; (iii) caso opte pelo menor espalhamento à direita (menor limitante superior), optará a árvore S^4 .

A figura 11 apresenta como foram obtidas as árvores do conjunto solução ao longo das cinco iterações.

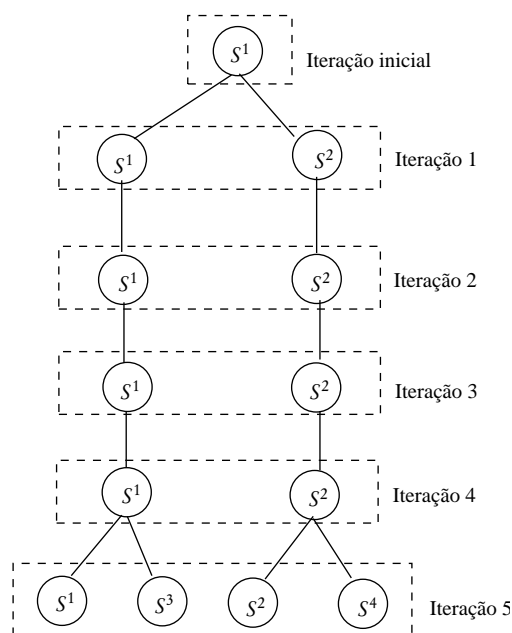


Figura 11. Obtenção das árvores geradoras ao longo das iterações – Instância 2

4.3 Instância 3

Nesta instância foi executada a Rede Óptica Italiana de Telecomunicações, utilizada em Takahashi e Yamakami [18] (Figura 12), com densidade de 17%, sendo os pesos das arestas as distâncias entre as cidades [36]. Considerando que os dados da

rede são reais (*crisp*), utilizou-se estes valores como os modais e um desvio de aproximadamente 10% para definir os espalhamentos à direita e à esquerda.

As tabelas 1 e 2 representam, respectivamente, os vértices, as arestas e seus pesos. Vale destacar que se trata de uma rede não direcionada, portanto as arestas da tabela 2 são válidas para ambos os sentidos.

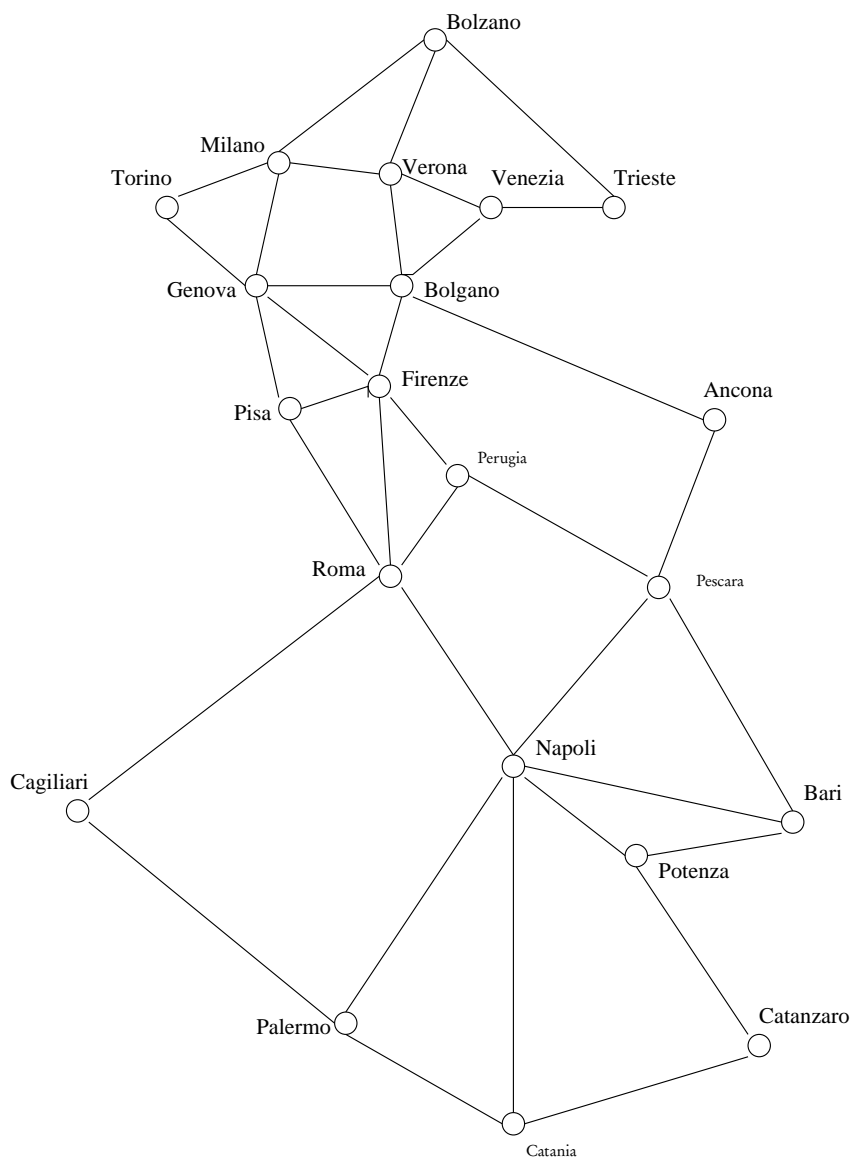


Figura 12. Rede 3 – Rede Óptica Italiana

Tabela 1. Denominação dos vértices da rede italiana

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

Tabela 2. Tabela dos arcos e custos da rede italiana

Aresta	Vértice 1	Vértice 2	Peso
1	1	2	(140 14 14)
2	1	3	(110 11 11)
3	1	4	(210 21 21)
4	2	3	(110 11 11)
5	3	5	(90 9 9)
6	4	5	(85 9 9)
7	2	6	(90 10 10)
8	2	7	(95 10 10)
9	3	8	(95 10 10)
10	5	8	(95 10 10)
11	6	7	(90 9 9)
12	7	8	(130 13 13)
13	7	9	(150 15 15)
14	7	10	(120 12 12)
15	8	9	(55 6 6)
16	8	11	(200 20 20)
17	9	10	(60 6 6)
18	9	12	(110 11 11)
19	9	13	(180 18 18)
20	10	13	(190 19 19)
21	11	14	(130 13 13)
22	12	14	(170 17 17)
23	12	13	(120 12 12)
24	13	17	(460 46 46)
25	13	15	(180 18 18)
26	14	15	(200 20 20)
27	14	16	(270 27 27)
28	15	16	(210 21 21)
29	15	18	(90 9 9)
30	15	20	(310 31 31)
31	15	21	(350 35 35)
32	16	18	(100 10 10)
33	17	20	(420 42 42)
34	18	19	(200 20 20)
35	19	21	(210 21 21)
36	20	21	(150 15 15)

Ao executar o algoritmo proposto foram encontradas duas árvores geradoras mínimas *fuzzy*, ambas com o peso (2660, 269, 269), sendo:

- Árvore 1, com as arestas: 2, 4, 5, 6, 8, 9, 11, 15, 17, 18, 21, 22, 23, 25, 29, 32, 33, 34, 35, 36; e
- Árvore 2, com as arestas: 2, 4, 5, 6, 8, 10, 11, 15, 17, 18, 21, 22, 23, 25, 29, 32, 33, 34, 35, 36.

Ao analisar o conjunto solução verifica-se que esta rede possui muitas arestas dominadas, o que ocasionou em um conjunto solução com apenas duas árvores geradoras. Além do mais, a diferença está nas arestas 9 e 10, que possuem os mesmos pesos.

Comparando os resultados obtidos com os de Takahashi [26] verifica-se que as arestas que possuem maior grau de pertinência nos algoritmos propostos por Takahashi [26] são as que fazem parte das soluções obtidas neste trabalho, o que podemos concluir que as soluções são idênticas.

5 Conclusões e trabalhos futuros

Neste trabalho foi proposto um algoritmo para o problema da árvore geradora mínima em redes com parâmetros incertos, utilizando a teoria dos conjuntos *fuzzy* para abordar as incertezas. Trata-se de uma adaptação do algoritmo clássico de Prim, seguindo as estratégias de alguns trabalhos de caminho mínimo *fuzzy* [4, 8], que utilizam índices de dominância para tratar as incertezas até a obtenção da solução final.

Embora o algoritmo proposto seja exato, o que pode dificultar sua aplicação em grandes redes, densas e com pouca dominância, este trabalho tem a vantagem de obter um conjunto solução de árvores geradoras, deixando a cargo do usuário, ou da aplicação, a escolha pela opção que interessar. Além do mais, vale destacar que os pesos das redes foram “defuzzificados”, utilizando três índices (valor modal, primeiro índice de Yager [37–39] e índice de Liou e Wang [40] – considerando $\lambda = 0,5$), e na sequência foi aplicado o algoritmo clássico de Prim, sendo que todas as árvores geradoras encontradas por meio destes índices fizeram parte das soluções encontradas no algoritmo proposto (Seção 4).

Vale destacar que os trabalhos da literatura que utilizam algoritmos genéticos podem resolver este problema em redes grandes, densas e com pouca dominância, ao contrário deste trabalho que é focado a redes menores e pouco densas.

Como trabalhos futuros pretende-se, além de propor uma ordenação ao conjunto solução do algoritmo proposto, verificar outras formas de ordenação e eliminação dos pesos, bem como abordar o tema via métodos heurísticos, principalmente quando se tratar de grafos densos e com pouca dominância entre as arestas.

Referências

- [1] ZADEH, L. A. Fuzzy sets. *Inform Control*, vol. 8, n. 3, p. 338-353, 1965.
- [2] ZADEH, L. A. Fuzzy algorithms. *Inform Control*, vol. 12, n. 2, p. 94-102, 1968.
- [3] ZADEH, L. A. Fuzzy sets as a basis for a theory of possibility. *Fuzzy Set Syst*, vol. 1, n. 1, p. 3-28, 1978.
- [4] OKADA, S.; SOPER, T. A shortest path problem on a network with fuzzy arc lengths. *Fuzzy Set Syst*, vol. 109, n. 1, p. 129-140, 2000.
- [5] BLUE, M.; BUSH, B.; PUCKETT, J. Unified approach to fuzzy graph problems. *Fuzzy Set Syst*, vol. 125, n. 3, p. 355-368, 2002.
- [6] OKADA, S. Fuzzy shortest path problems incorporating interactivity among paths. *Fuzzy Set Syst*, vol. 142, n. 3, p. 335-357, 2004.
- [7] NAYEEM, S. M. A.; PAL, M. Shortest path problem on a network with imprecise edge weight. *Fuzzy Optim Decis Ma*, vol. 4, n. 4, p. 293-312, 2005.
- [8] HERNANDES, F.; LAMATA, M. T.; VERDEGAY, J. L.; YAMAKAMI, A. The shortest path problem on networks with fuzzy parameters. *Fuzzy Set Syst*, vol. 158, n. 14, p. 1561-1570, 2007.
- [9] KHORRAM, E.; KESHAVARZ, E. A fuzzy shortest path with the highest reliability. *J Comput Appl Math*, vol. 230, n. 1, p. 204-212, 2009.

- [10] DENG, Y.; CHEN, Y.; ZHANG, Y.; MAHADEVAN, S. Fuzzy Dijkstra algorithm for shortest path problem under uncertain environment. *Appl Soft Comput*, vol. 12, n. 3, p. 1231-1237, 2012.
- [11] ELIZABETH, S.; SUJATHA, L. Fuzzy shortest path problem based on level λ -triangular fuzzy numbers. *Adv Fuzzy Syst*, n. 1, p. 1-6, 2012.
- [12] KIM, K.; ROUSH, F. Fuzzy flows on networks. *Fuzzy Set Syst*, vol. 8, n. 1, p. 35-38, 1982.
- [13] CHANAS, S.; KOLODZIEJCZYK, W. Integer flows in network with fuzzy capacity constraints. *Fuzzy Set Syst*, vol. 16, n. 1, p. 17-31, 1986.
- [14] HERNANDES, F.; LAMATA, M. T.; TAKAHASHI, M. T.; YAMAKAMI, A.; VERDEGAY, J. L. An algorithm for the fuzzy maximum flow problem. *IEEE Int Conf Fuzzy*, vol. 16, n. 1, p. 776-781, 2007.
- [15] KUMAR, A.; KAUR, M. An improved algorithm for solving fuzzy maximal flow problem. *International Journal of Applied Science and Engineering*, vol. 10, n. 1, p. 19-27, 2012.
- [16] SHIH, H. -S.; LEE, E. S. Fuzzy multi-level minimum cost flow problems. *Fuzzy Set Syst*, vol. 107, n. 2, p. 159-176, 199.
- [17] TAKAHASHI, M. T.; YAMAKAMI, A. Um estudo sobre árvore geradora mínima com estrutura do grafo fuzzy. *XXXVI Simpósio Brasileiro de Pesquisa Operacional*, vol. 1, p. 2278-2285, 2004.
- [18] TAKAHASHI, M. T.; YAMAKAMI, A. Um estudo sobre árvore geradora mínima com parâmetro fuzzy. *XXXV Simpósio Brasileiro de Pesquisa Operacional*, vol. 1, p. 2217-2227, 2003.
- [19] MOHANTY, S. P.; BISUAL, S.; PRADHAN, G. Minimum spanning tree in fuzzy weighted rough graph. *Int J Adv Eng Res Dev*, vol. 1, n. 10, p. 23-28, 2012.
- [20] ZHOU, J.; WANG, Q.; ZHANG, X. The inverse spanning tree of a fuzzy graph based on credibility measure. *J Commun*, vol. 8, n. 9, p. 566-571, 2013.

- [21] NAYEEM, S. M. A.; PAL, M. Diameter constrained fuzzy minimum spanning tree problem. *Int J Comput Int Sys*, vol. 6, n. 6, p. 1040-1051, 2013.
- [22] RAMIRES, J. Extensiones del problema de coloración de grafos. Tese de Doutorado. Universidad Complutense de Madrid, Madrid, 2001.
- [23] YÁÑEZ, J.; RAMÍREZ, J. The robust coloring problem. *Eur J Oper Res*, vol. 148, n. 3, p. 546-558, 2003.
- [24] MUÑOZ, S.; ORTUÑO, T. RAMÍREZ, J.; YÁÑEZ, J. Coloración de grafos difusos, aplicación a problemas de planificación de exámenes. *27 Congreso Nacional de Estadística e Investigación Operativa*, vol. 1, p. 3512-3518, 2003.
- [25] MUÑOZ, S.; ORTUÑO, T. RAMÍREZ, J.; YÁÑEZ, J. Coloring fuzzy graphs. *Omega-Int J Manage S*, vol. 33, n. 3, p. 211-22, 2005.
- [26] TAKAHASHI, M. T. Contribuições ao estudo de grafos *fuzzy*: teoria e algoritmos. Tese de Doutorado. Universidade Estadual de Campinas, Campinas, 2004.
- [27] AHUJA, R. K.; MAGNANTI, T. L.; ORLIN, J. B. Network flows: theory, algorithms and application. Prentice Hall. New Jersey, 1993.
- [28] DELGADO, M.; VERDEGAY, J. L.; VILA, M. A. On fuzzy tree definition. *Eur J Oper Res*, vol. 22, n. 2, p. 243-249, 1985.
- [29] CHUNDE, Y. On the optimization problem of spanning tree in fuzzy network. *The Journal of China Universities of Posts and Telecommunications*, vol. 3, n. 2, p. 22-25, 1996.
- [30] GAO, J.; LU, M. Fuzzy quadratic minimum spanning tree problem. *Appl Math Comput*, vol. 164, n. 3, p. 773-788, 2005.
- [31] JANIÁK, A.; KASPERKI, A. The minimum spanning tree problem with fuzzy costs. *Fuzzy Optim Decis Ma*, vol. 7, n. 2, p. 105-118, 2008.
- [32] YAO, J. S.; LIN, F. T. Fuzzy shortest path network problems with uncertain edge weight. *J Inf Sci Eng*, vol. 19, n. 2, p. 329-351, 2003.

- [33] GONDRAN, M.; MINOUX, M. Graphs and algorithms. John Wiley & Sons. New York, 1984.
- [34] DUBOIS, D.; PRADE, H. Fuzzy sets and systems: teory and applications. Academic Press. New York, 1980.
- [35] PEDRYCZ, W.; GOMIDE, F. An introduction to fuzzy sets: analysis and design. MIT Press. London, 1998.
- [36] ALI, M.; RAMAMURTHY, B.; DEOGUN, J. Routing and wavelength assignment with power considerations in optical networks. *Comput Netw*, vol. 32, p. 539-555, 2000.
- [37] YAGER, R. R. Ranking fuzzy subsets over the unit interval. *IEEE Decis Contr P*, vol. 17, p. 1435-1437, 1978.
- [38] YAGER, R. R. On Choosing between fuzzy subsets. *Kybernets*, vol. 9, p. 151-154, 1980.
- [39] YAGER, R. R. A procedure for ordering fuzzy subsets of the unit interval. *Inform Sciences*, vol. 24, n. 2, p. 143-161, 1981.
- [40] LIOU, T. -S.; WANG, M. -J. Ranking fuzzy numbers with integral interval. *Fuzzy Set Syst*, vol. 50, n. 2, p. 247-255, 1992.