

O PROBLEMA DE CAMINHO MÍNIMO COM INCERTEZAS E RESTRIÇÕES DE TEMPO

Fábio Hernandes*

Departamento de Ciência da Computação (DECOMP)
Universidade Estadual do Centro-Oeste (UNICENTRO)
Guarapuava – PR
hernandes@unicentro.br; hernandes.fabio@gmail.com

Lilian Berton

Departamento de Ciência da Computação (DECOMP)
Departamento de Matemática (DEMAT)
Universidade Estadual do Centro-Oeste (UNICENTRO)
Guarapuava – PR
lilianberton@yahoo.com.br

Maria José de Paula Castanho

Departamento de Matemática (DEMAT)
Universidade Estadual do Centro-Oeste (UNICENTRO)
Guarapuava – PR
zeza@unicentro.br

* *Corresponding author* / autor para quem as correspondências devem ser encaminhadas

Recebido em 08/2008; aceito em 04/2009
Received August 2008; accepted April 2009

Resumo

O problema de caminho mínimo em grafos com parâmetros incertos é um importante problema da programação matemática, visto que possui aplicações nas mais diversas áreas da Computação e da Engenharia, como: redes de computadores, telecomunicações, transportes, manufaturas, dentre outros. Porém, devido à sua alta complexidade computacional, poucos são os algoritmos existentes na literatura. Neste trabalho é proposto um algoritmo, baseado no algoritmo de Okada & Soper, que utiliza dois parâmetros incertos nos arcos, custo e tempo, com restrições de tempo nos nós. As incertezas são abordadas utilizando a teoria dos conjuntos *fuzzy*.

Palavras-chave: teoria dos grafos; problema de caminho mínimo; teoria dos conjuntos *fuzzy*.

Abstract

The shortest path problem in graphs with uncertainties in the parameters is an important problem in the mathematical programming, since it has a wide range of applications in different areas of Computation and Engineering, such as: computer networks, telecommunications, transportation, manufacturing, etc. However, due to its high computational complexity, there are few algorithms in the literature. In this paper is proposed an algorithm, based on Okada and Soper algorithm, which uses two uncertainty parameters, cost and time, with time restrictions in the nodes. The uncertainties are discussed using the fuzzy set theory.

Keywords: graphs theory; shortest path problem; fuzzy set theory.

1. Introdução

Incertezas, imprecisões, e ambiguidades são fatores que nem sempre podem ser ignorados na resolução de problemas. Introduzida por Zadeh nas décadas de 1960 e 1970 (Zadeh, 1965, 1968, 1978), a teoria de conjuntos *fuzzy* é uma ferramenta frequentemente utilizada na modelagem desses fatores. Atualmente, diversos trabalhos nas áreas da Engenharia e da Computação utilizam esta teoria (Moderson & Nair, 2000; Takahashi & Yamakami, 2004; Hernandez, 2007). Problemas de controle (Shaw & Simões, 1998), sistemas de suporte à decisão (Bellman & Zadeh, 1970), reconhecimento de padrões (Bezdek, 1981), e programação matemática (Malik & Moderson, 2001; Hernandez, 2007) constituem uma pequena amostra do espectro de suas aplicações.

Na programação matemática a área de grafos com incertezas é uma das mais estudadas, visto que grande quantidade de problemas pode ser representada na forma de redes: caminho mínimo (Blue *et al.*, 2002; Hernandez & Takahashi, 2004; Hernandez, 2007; Hernandez *et al.*, 2007; Klein, 1991; Lin & Chern, 1993; Okada & Soper, 2000; Okada, 2004); fluxo máximo (Chanas *et al.*, 1995; Hernandez *et al.*, 2006); fluxo de custo mínimo (Kim & Roush, 1982; Shih & Lee, 1999; Takahashi, 2004); dentre outras.

Embora a área de grafos tenha um vasto campo de aplicações, o estudo de grafos *fuzzy* ainda se encontra na fase inicial. Além do mais, a maioria destes problemas possui alta complexidade computacional, tornando difícil suas soluções. Em muitas aplicações do problema de caminho mínimo as restrições de custo e tempo não são rígidas, o que propicia o uso da teoria dos conjuntos *fuzzy* e o problema é, então, denominado “Problema de Caminho Mínimo *Fuzzy*” (PCM-*fuzzy*). Neste trabalho este tema é abordado, sendo que as incertezas presentes estão em dois parâmetros dos arcos e nas restrições de tempo nos nós. As incertezas dos arcos estão nos custos e nos tempos necessários para o fluxo percorrer aquele arco. As restrições de tempo estão em cada nó i , sendo que a soma dos tempos dos arcos dos caminhos entre o nó 1 e i não pode ultrapassar a restrição de tempo do referido nó i .

Dentre os principais trabalhos de caminho mínimo *fuzzy* encontrados na literatura destacam-se: Dubois & Prade (1980), Klein (1991), Okada & Soper (2000), Blue *et al.* (2002), Okada (2004), Chuang & Kung (2005), Nayeem & Pal (2005) e Hernandez *et al.* (2007). O trabalho de Dubois & Prade (1980) está entre um dos primeiros da literatura, onde é proposta a simples extensão dos algoritmos clássicos de Floyd e de Ford-Moore-Bellman. Porém, verificou-se que ambos os algoritmos podem retornar comprimentos sem um caminho associado. Em 1991, Klein contorna a dificuldade encontrada por Dubois & Prade com o uso de dominância de conjuntos *fuzzy* e propõe um algoritmo para determinar o menor caminho. O trabalho de Okada & Soper (2000) destaca-se pela obtenção da solução final, que é composta por um conjunto de caminhos não-dominados, sendo que cada elemento da solução é um caminho com certo grau de pertinência de ser o mínimo. Blue *et al.* (2002) apresentam uma taxonomia para grafos e propõem alguns algoritmos para os principais problemas. No algoritmo de caminho mínimo é determinado um valor de corte para limitar o número de caminhos a serem analisados; em seguida aplica-se uma versão modificada do algoritmo clássico dos k -caminhos mínimos, proposto por Eppstein (1994), a fim de se determinar os caminhos com algum grau de possibilidade de ser o mínimo. Okada (2004) estudou a interatividade entre os caminhos *fuzzy* e desenvolveu um algoritmo utilizando teoria de possibilidade, no qual introduz o conceito de grau de possibilidade de um arco pertencer a um caminho mínimo. Já Chuang & Kung (2005) propõem uma heurística para encontrar o comprimento do caminho mínimo, além do conceito de grau de similaridade,

cuja finalidade é verificar, pela heurística proposta, a similaridade entre o comprimento mínimo encontrado, e os comprimentos de todos os possíveis caminhos. No trabalho de Nayeem & Pal (2005) é proposto um algoritmo baseado no índice de aceitabilidade de Sengupta & Pal (2000), em que o usuário, de acordo com seu ponto de vista (otimista/pessimista), deve escolher o melhor caminho. Porém, esta escolha só ocorre quando na solução final houver mais de um caminho entre dois nós. Um dos trabalhos mais recentes da literatura, Hernandez *et al.* (2007), propõe um algoritmo, baseado no algoritmo clássico de Ford-Moore-Bellman, que trabalha com as incertezas até a obtenção da solução final. Este trabalho também determina um conjunto solução de caminhos não-dominados, semelhante ao de Okada & Soper (2000), com a vantagem de poder ser aplicado em redes com parâmetros negativos.

Analisando os trabalhos citados anteriormente verifica-se que apresentam algumas desvantagens, tais como: alguns encontram custos sem caminhos associados; outros utilizam índices de “defuzzificação” para transformar o problema incerto em um problema clássico ocorrendo perda de informações. Os trabalhos mais recentes apresentam um conjunto solução de caminhos, sem apresentar ao usuário uma ordenação dos mesmos. Considerando o PCM-*fuzzy* com restrições de tempo, o único trabalho da literatura é o de Hernandez & Takahashi (2004), baseado no algoritmo de Ford-Moore-Bellman (Bellman, 1958). Porém, este trabalho apresenta como solução final um conjunto de caminhos não-dominados sem ordenação, deixando a cargo do usuário a decisão por um dos caminhos do conjunto. O algoritmo aqui proposto apresenta o conjunto solução ordenado.

Este trabalho está organizado da seguinte forma: na Seção 2 são apresentados alguns conceitos da teoria dos conjuntos *fuzzy*. A Seção 3 aborda o problema do caminho mínimo com incertezas apresentando o algoritmo proposto. Na Seção 4 estão os resultados computacionais. Conclusões e propostas futuras são discutidas na Seção 5.

2. Teoria dos Conjuntos *Fuzzy*

A teoria dos conjuntos *fuzzy* foi introduzida nas décadas de 60 e 70 pelo matemático Lotfi Asker Zadeh como uma teoria matemática aplicada a conceitos vagos, com a intenção de dar um tratamento matemático a termos linguísticos subjetivos.

A principal vantagem de se usar esta teoria está na capacidade adquirida de representar e manipular informações e conhecimentos expressos numa linguagem natural, como, por exemplo, “aproximadamente” e “em torno de”.

Nesta seção são apresentados alguns conceitos utilizados neste trabalho (Bassanezi & Barros, 2006; Pedrycz & Gomide, 1998).

2.1 Conceitos Básicos

Definição 1: Seja U um conjunto (clássico). Um conjunto *fuzzy* F de U é caracterizado por uma função $\mu_F : U \rightarrow [0,1]$, chamada *função de pertinência* do conjunto *fuzzy*. O valor $\mu_F(x) \in [0,1]$ indica o grau com que o elemento x de U pertence ao conjunto *fuzzy* F ; $\mu_F(x) = 0$ e $\mu_F(x) = 1$ indicam, respectivamente, a não pertinência e a pertinência completa de x ao conjunto *fuzzy* F .

Do ponto de vista formal, a definição de conjuntos *fuzzy* foi obtida simplesmente ampliando-se o contradomínio da função característica clássica, que é o conjunto $\{0,1\}$, para o intervalo $[0,1]$. Nesse sentido, pode-se dizer que um conjunto clássico é um caso particular de conjunto *fuzzy* cuja função de pertinência μ_F é a sua função característica X . Um conjunto clássico, na linguagem *fuzzy*, costuma ser denominado *conjunto crisp*.

O subconjunto clássico de U definido por $\text{supp}(A) = \{x \in U : \mu_A(x) > 0\}$, denominado suporte de F , é de fundamental importância na interrelação entre as teorias de conjuntos clássica e *fuzzy*.

Definição 2: Seja A um conjunto *fuzzy* de U e $\alpha \in [0,1]$. Um α -corte de A é o conjunto clássico de U definido por $[A]^\alpha = \{x \in U : \mu_A(x) \geq \alpha\}$ para $0 \leq \alpha \leq 1$.

Se $[A]^\alpha = \{x \in U : \mu_A(x) > \alpha\}$, $[A]^\alpha$ é dito α -corte forte.

O nível zero de um conjunto *fuzzy* A é definido como sendo o menor conjunto (clássico) fechado de U que contém o conjunto suporte de A . Numa linguagem matemática, $[A]^0$ é o fecho do suporte de A e é indicado por $\text{supp}(A)$.

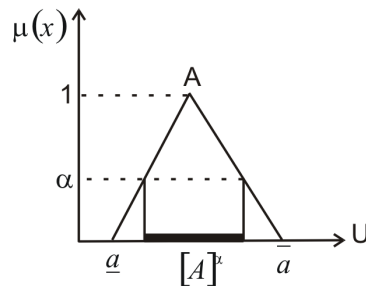


Figura 1 – Exemplo de α -corte.

2.2 Operações entre Conjuntos Fuzzy

Sejam A e B conjuntos *fuzzy* de U , com funções de pertinência indicadas por $\mu_A(x)$ e $\mu_B(x)$, respectivamente. As operações de união, intersecção e complementar são definidas da seguinte forma (Figura 2):

- A união entre A e B é o conjunto *fuzzy* de U cuja função de pertinência é dada por: $\mu_{(A \cup B)}(x) = \max\{\mu_A(x), \mu_B(x)\}, \forall x \in U$.
- A intersecção entre A e B é o conjunto *fuzzy* de U cuja função de pertinência é dada por: $\mu_{(A \cap B)}(x) = \min\{\mu_A(x), \mu_B(x)\}, \forall x \in U$.
- O complementar de A é o conjunto *fuzzy* A' de U cuja função de pertinência é dada por: $\mu_{A'}(x) = 1 - \mu_A(x), \forall x \in U$.

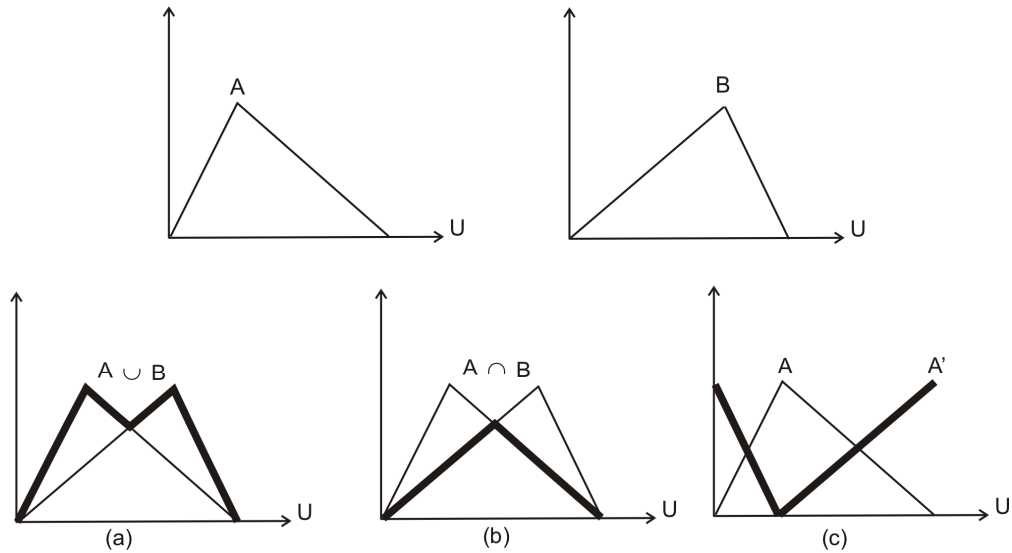


Figura 2 – Operações entre conjuntos *fuzzy*: (a) união; (b) intersecção; (c) complementar.

2.3 Números *Fuzzy*

Existem vários tipos de números *fuzzy* (Pedrycz & Gomide, 1998), entretanto, neste trabalho são utilizados apenas números *fuzzy* triangulares.

Definição 3: *Valor modal* é o valor $x \in [\underline{a}, \bar{a}]$ para o qual a função de pertinência tem valor máximo.

Definição 4: Um *número fuzzy triangular* pode ser definido como $\tilde{a} = (\underline{a}, a, \bar{a})$, tendo sua função de pertinência, $\mu_{\tilde{a}}(x) : \mathfrak{R} \rightarrow [0, 1]$ (Figura 3), dada por:

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

tal que:

- \underline{a} : limitante inferior;
- a : valor modal; e
- \bar{a} : limitante superior.

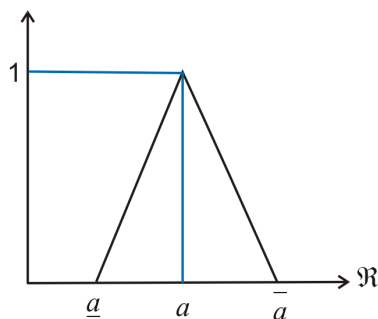


Figura 3 – Número fuzzy triangular.

Definição 5: Sejam \tilde{a} e \tilde{b} dois números fuzzy triangulares, $\tilde{a} = (\underline{a}, a, \bar{a})$ e $\tilde{b} = (\underline{b}, b, \bar{b})$, então a soma fuzzy é denotada por:

$$\tilde{a} \oplus \tilde{b} = (\underline{a}, a, \bar{a}) \oplus (\underline{b}, b, \bar{b}) = (\underline{a} + \underline{b}, a + b, \bar{a} + \bar{b})$$

Definição 6: Sejam $\tilde{a} = (\underline{a}, a, \bar{a})$ e $\tilde{b} = (\underline{b}, b, \bar{b})$ dois números fuzzy triangulares, então $\tilde{a} \prec \tilde{b}$ (\tilde{a} domina \tilde{b}) se, e somente se, $\underline{a} \leq \underline{b}$, $a \leq b$, $\bar{a} \leq \bar{b}$ e $\tilde{a} \neq \tilde{b}$.

Definição 7: Um valor $\tilde{a} = (\underline{a}, a, \bar{a})$ é lexicograficamente menor do que $\tilde{b} = (\underline{b}, b, \bar{b})$ se pelo menos uma das condições for satisfeita:

$$\underline{a} < \underline{b}; \underline{a} = \underline{b} \text{ e } a < b; \underline{a} = \underline{b}, a = b \text{ e } \bar{a} < \bar{b}.$$

2.4 Teoria de Possibilidade

Seja um grafo $G=(V,A)$ com custos $\tilde{c} \in \mathfrak{R}^n$, $\tilde{c} = \{\tilde{c}_{ij}\}$, associados aos seus arcos. Sejam dois subgrafos T^1 e T^2 , $T^1 \neq T^2$. Pode-se dizer que T^1 tem um grau de possibilidade de ser menor do que T^2 dado por (Okada, 2004):

$$\tilde{w} = Poss \left(\sum_{ij \in T^1} \tilde{c}_{ij} \leq \sum_{ij \in T^2} \tilde{c}_{ij} \right) = \sup_{u < v} \min \{ \mu_{T^1}(u), \mu_{T^2}(v) \}$$

3. O Problema do Caminho Mínimo com Incertezas

O problema do caminho mínimo está associado a várias aplicações nas áreas de roteamento, manufatura, fluxo em redes, dentre outras. Seu principal objetivo é minimizar os esforços de busca.

Sendo u e v dois vértices de um grafo $G=(V,A)$, o caminho mínimo entre u e v é uma sequência de arcos que, passando por nós distintos, une u a v de forma a acumular o menor comprimento (distância). Para que haja um caminho mínimo entre dois arcos u e v é

necessário que exista uma conexão entre eles, isto é, v seja sucessor de u (Goldberg & Luna, 2000). Na maioria dos casos, existe mais de um caminho entre dois nós específicos; com isso, o problema do caminho mínimo consiste em encontrar o caminho com menor custo dentre todos os possíveis.

Na teoria dos grafos clássica, como o peso (custo) de cada arco é um número real (*crisp*), existem algoritmos eficientes para resolver este problema (Goldberg & Luna, 2000). Porém, em muitas aplicações reais estes parâmetros (custos, capacidades, dentre outros) não são naturalmente precisos, muitas vezes, por falta de informações.

O algoritmo aqui proposto aborda incertezas nos arcos, além de incluir restrições de tempo nos nós. É composto de duas fases: na primeira é realizada uma adaptação do algoritmo de Okada & Soper (2000) para a detecção do conjunto solução de caminhos não-dominados; na segunda é feita a distribuição do fluxo ao longo dos arcos dos caminhos não-dominados.

3.1 Abordagem de Okada & Soper

O algoritmo proposto por Okada & Soper (2000) é uma modificação do algoritmo de Martin (1984) e foi desenvolvido com base no método de múltiplas etiquetas de Hansen (1980). Este método pode ser considerado como uma generalização do algoritmo de Dijkstra, que resolve o problema do caminho mínimo clássico.

Neste algoritmo os comprimentos dos arcos são dados por números *fuzzy* triangulares positivos. Como o custo do caminho é dado pela soma dos custos dos arcos, este também é representado por um número *fuzzy*. A solução, baseada no conceito de dominância, contém todos os caminhos não-dominados de um nó s a um nó t (Definição 6).

Como a relação de dominância pode apresentar mais de um caminho não-dominado entre dois nós, cada caminho recebe um rótulo (etiqueta) para que este seja composto no passo final do algoritmo. As etiquetas são divididas em dois conjuntos: o conjunto das etiquetas temporárias e o das etiquetas permanentes. Enquanto uma etiqueta temporária pode ser excluída, ou mudada para uma etiqueta permanente, uma permanente permanece inalterada.

Para selecionar tanto uma etiqueta temporária como uma permanente, no algoritmo de Dijkstra é escolhida aquela com a menor distância (custo); já no algoritmo de Okada & Soper é utilizada uma relação de ordem lexicográfica.

3.2 O Problema do Caminho Mínimo com Incertezas e Restrições de Tempo

Neste problema a rede a ser considerada, além de possuir os dois parâmetros incertos nos arcos (custo e tempo), também possui uma restrição de tempo em cada um dos nós, isto é, a soma dos tempos dos arcos que pertencem ao caminho, entre os nós 1 e i , não pode ultrapassar o tempo máximo estipulado na restrição do referido nó i . Considerando que os parâmetros dos arcos e as restrições dos nós são números *fuzzy*, as verificações das restrições são satisfeitas utilizando a medida de possibilidade, definida da seguinte forma:

Definição 8: Seja $\tilde{t}_{i(1,j)}$ o tempo da i -ésima etiqueta do caminho entre os nós 1 e j , e seja Rt_j a restrição de tempo do nó j , então a restrição de tempo é satisfeita se $Pt = Poss(\tilde{t}_{i(1,j)} \leq Rt_j) > \beta, \beta \in [0, 1]$.

O valor de β ($\beta \in [0, 1]$) fica a cargo do usuário, pois depende da aplicação em questão.

Como na solução final é encontrado um conjunto de caminhos não-dominados, em algumas situações haverá mais de um caminho entre um par de nós, com isso é necessário ordená-los.

A ordenação em relação ao tempo (e ao custo dos caminhos) é feita usando a medida de possibilidade (Definição 8). Maior medida de possibilidade indica melhor caminho em relação ao tempo, P_t e em relação ao custo P_c . Porém, há casos em que gera indecisões sobre qual caminho escolher: *um caminho barato e relativamente lento ou um caminho relativamente rápido e mais caro?*

Para resolver esta indecisão foi proposta uma combinação convexa entre estes parâmetros incertos ($\alpha \cdot P_c + (1 - \alpha) \cdot P_t$), tal que $\alpha \in [0, 1]$. Esta proposta é detalhada nos resultados computacionais.

3.3 Algoritmo Proposto

O objetivo deste algoritmo é encontrar todos os caminhos não-dominados entre os nós 1 e i ($i=2, 3, \dots, n$) que satisfaçam as restrições de tempo entre os nós. Os passos 1, 2 e 3 do algoritmo correspondem à adaptação do algoritmo de Okada & Soper.

3.3.1 Algoritmo

Informações sobre o algoritmo:

N : conjunto dos nós.

A : conjunto dos arcos.

T : conjunto dos caminhos temporários, sendo cada elemento (i, k) o caminho do nó 1 a i com a etiqueta k .

P : conjunto dos caminhos permanentes.

$\tilde{d}_{(sj)}$: custo do arco (s, j) .

$\tilde{t}_{(sj)}$: tempo do arco (s, j) .

$\tilde{d}_{i(1j)}$: custo da i -ésima etiqueta do caminho entre os nós 1 e j .

$\tilde{t}_{i(1j)}$: tempo da i -ésima etiqueta do caminho entre os nós 1 e j .

Rt_j : restrição de tempo do nó j .

$\tilde{d}_{i(1j)} < \tilde{d}_{u(1j)}$: o custo do caminho do nó 1 ao nó j dado pela etiqueta i domina o custo dado pela etiqueta u (Definição 6).

Algoritmo

PASSO 0: Início

- i) Associe a etiqueta $[\tilde{d}_{i(1j)}, (i, j)]$ para os nós $j=I$, tal que $\tilde{d} = (0, 0, 0)$.
- ii) $T \leftarrow (1,1)$ e $P \leftarrow \phi$.

PASSO 1: Seleção das etiquetas

- i) Se $T = \phi$, vá ao Passo 4. Caso contrário, entre todas as etiquetas temporárias determine qual é a lexicograficamente menor. Seja a i -ésima etiqueta associada ao nó j .
- ii) Descartar (i, j) de T e acrescentar ao conjunto P como segue: $T \leftarrow T - \{(i, j)\}$ e $P \leftarrow P \cup \{(i, j)\}$.

PASSO 2: Análise de etiquetas e verificação de dominância

Para cada nó $j \in N$ tal que $(i, j) \in A$, execute:

- i) Determine o custo e o tempo do caminho *fuzzy* como segue: $\tilde{d}_{i(1j)} = \tilde{d}_{k(1l)} \oplus \tilde{d}_{(lj)}$ e $\tilde{t}_{i(1j)} = \tilde{t}_{k(1l)} \oplus \tilde{t}_{(lj)}$.
- ii) Seja $[\tilde{d}_{i(1j)}, (i, j)]$ a i -ésima etiqueta do nó j . Atualize o conjunto T como segue: $T \leftarrow T \cup (i, j)$.
- iii) A verificação da dominância entre todas as etiquetas do nó j é executada como segue:
 - Para cada $\tilde{d}_{u(1j)}$, tal que $(u, j) \in T \cup P$, se $\tilde{d}_{i(1j)} \succ \tilde{d}_{u(1j)}$ então delete a i -ésima etiqueta do nó j , descarte o correspondente (i, j) de T e pare o procedimento iii) neste passo.
 - Para cada $\tilde{d}_{u(1j)}$, tal que $(u, j) \in T \cup P$, se $\tilde{d}_{i(1j)} \prec \tilde{d}_{u(1j)}$ então delete a u -ésima etiqueta do nó j e descarte o correspondente (u, j) de T .
 - Caso contrário, não faça nada.
- iv) Para todos os caminhos não-dominados do nó j que estão na lista dos temporários fazer:

$$Pt = Poss(\tilde{t}_{i(1j)} \leq Rt_j)$$

Se $Pt < \beta$ então o caminho é eliminado.

PASSO 3: Retorne ao Passo 1.

PASSO 4: Composição dos caminhos não-dominados

Encontre os caminhos não-dominados de I para j e caso tenha mais de um caminho não dominado entre um par de vértices faça:

- Calcule a possibilidade de cada um desses caminhos ser o menor.

$$Pc_i = Poss(\tilde{d}_{i(1,j)} \leq \tilde{d}_{u(1,j)})$$

- $Pc = \min\{Pc_{ij}\}$

PASSO 5: Ordenação dos caminhos finais

Após o cálculo da possibilidade de cada caminho ser o menor, faça:

- i) Ordem $(i) \leftarrow 0$;
- ii) Enquanto todos os caminhos não-dominados não estiverem ordenados faça:
 1. Se um caminho k possuir os valores de Pt e Pc maiores que os valores dos demais caminhos, então:
 - Ordem $(i) \leftarrow$ caminho k ;
 - $i \leftarrow i+1$;
 - Descarte o *caminho* k da lista dos não ordenados e volte ao início da lista.
 2. Senão, os caminhos não-dominados são ordenados da seguinte forma:

Para cada caminho não-ordenado faça:

$$\text{Ordenação} = (\alpha \cdot Pc + (1 - \alpha) \cdot Pt)$$

tal que $\alpha \in [0,1]$, cabendo ao usuário a decisão sobre qual dos dois parâmetros é mais importante, custo ou tempo.

3. Após calculada a ordenação faça:
 - i) Encontre o caminho k que possui maior valor de ordenação
 - Ordem $(i) \leftarrow$ caminho k ;
 - $i \leftarrow i+1$;
 - Descarte o *caminho* k da lista dos não ordenados e volte ao início desta etapa 3.

PASSO 6: FIM.

3.3.2 Complexidade Computacional

Seja V_j o número de etiquetas permanentes do nó j e $V_{max} = \max\{V_1, V_2, \dots, V_r\}$. Sendo r o número de nós do grafo. No Passo 2 são efetuadas no máximo rV_{max} somas para calcular o custo, rV_{max} somas para o cálculo do tempo e rV_{max}^2 comparações para dominância, possuindo portanto complexidade $O(rV_{max}^2) = O(rV_{max} + rV_{max} + rV_{max}^2)$. Considerando que o algoritmo proposto é uma adaptação de Dijkstra, cuja complexidade é $O(r^2)$, e que os demais passos não possuem complexidade maior que o Passo 2, tem-se que a complexidade do algoritmo proposto é limitada por $O(r^2(rV_{max}^2)) = O(r^3V_{max}^2)$ (Okada & Soper, 2000).

4. Resultados Computacionais

O algoritmo proposto foi implementado em *Java Standard Edition*, versão 5.0, e a ferramenta *BlueJ*, versão 2.1.3, sendo executado em duas redes (Figuras 4 e 5). Nos exemplos trabalhados os valores considerados para β foram iguais a zero, ou seja, se a medida de possibilidade (Definição 8) tiver valor diferente do nulo significa que a restrição do referido nó foi satisfeita. Porém, como dito anteriormente, cabe ao usuário decidir sobre o valor de β .

Exemplo 1.

Este exemplo trata de uma pequena rede (Figura 4), sendo a meta comentar mais detalhadamente os caminhos não-dominados. Os números *fuzzy* triangulares que estão nos nós da respectiva figura representam as restrições de tempo.

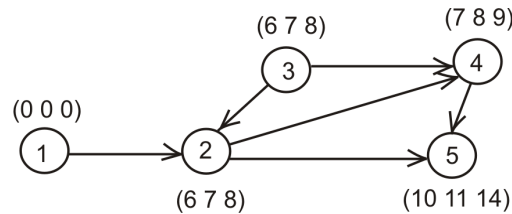


Figura 4 – Exemplo de uma pequena rede.

A Tabela 1 apresenta os parâmetros incertos dos arcos da Figura 4.

Tabela 1 – Dados da rede da Figura 4.

Arco	Origem	Destino	Custo	Tempo
1	1	2	(20 23 26)	(1 2 3)
2	2	3	(5 7 9)	(2 3 4)
3	2	4	(10 12 14)	(4 5 6)
4	2	5	(22 28 32)	(1 2 3)
5	3	4	(10 12 15)	(2 3 4)
6	4	5	(13 15 17)	(4 5 6)

Aplicando o algoritmo proposto foram obtidos os seguintes resultados:

- Entre os nós 1 e i ($i = 2,3,4$) foi obtido apenas um caminho não-dominado que satisfaz a respectiva restrição de tempo do respectivo nó, não sendo necessário ordená-lo.
- Os caminhos detectados entre os nós 1 e 5 foram:
 - Caminho 1: $1 \rightarrow 2 \rightarrow 5$, com custo = (42 51 58), tempo = (2 4 6); $P_t = 1,0$.
 - Caminho 2: $1 \rightarrow 2 \rightarrow 4 \rightarrow 5$, com custo = (43 50 57), tempo = (9 12 15); $P_t = 0,8333$.

Analisando os dois caminhos não-dominados entre os nós 1 e 5, a possibilidade do custo do caminho 1 ser menor que a do caminho 2 é calculada por:

$$Pc = Poss(\tilde{d}_{1(15)} \leq \tilde{d}_{2(15)}) \rightarrow Pc = Poss((42 \ 51 \ 58) \leq (43 \ 50 \ 57)) \rightarrow Pc = 0,5$$

A possibilidade do custo do caminho 2 ser menor que a do caminho 1 é calculada por:

$$Pc = Poss(\tilde{d}_{2(15)} \leq \tilde{d}_{1(15)}) \rightarrow Pc = Poss((43 \ 50 \ 57) \leq (42 \ 51 \ 58)) \rightarrow Pc = 1,0$$

Com os cálculos das medidas de possibilidade verificou-se que o Caminho 1 possui $Pt=1$, maior do que a do Caminho 2 ($Pt=0,833$), porém o valor de seu Pc é menor do que a do Caminho 2. Logo, caberá ao usuário decidir qual destes dois caminhos escolher: se o mesmo priorizar o tempo, o valor de α deverá ser “pequeno” e caso priorize o custo, o valor de α deverá ser oposto.

Por exemplo:

Seja $\alpha=0,7$ (priorizando o custo), os resultados são:

- Caminho 1 = $(\alpha.Pc + (1-\alpha).Pt) \rightarrow$ caminho 1 = $(0,7(0,5) + (1-0,7)(1,0)) \rightarrow$ caminho 1 = 0,65;
- Caminho 2 = $(\alpha.Pc + (1-\alpha).Pt) \rightarrow$ caminho 2 = $(0,7(1,0) + (1-0,7)(0,8333)) \rightarrow$ caminho 2 = 0,95.

Logo, os caminhos ficam ordenados da seguinte forma: caminho 2, caminho 1.

Considerando $\alpha=0,2$ (priorizando o tempo), os resultados são:

- Caminho 1 = $(\alpha.Pc + (1-\alpha).Pt) \rightarrow$ caminho 1 = $(0,2(0,5) + (1-0,2)(1,0)) \rightarrow$ caminho 1 = 0,9;
- Caminho 2 = $(\alpha.Pc + (1-\alpha).Pt) \rightarrow$ caminho 2 = $(0,2(1,0) + (1-0,2)(0,8333)) \rightarrow$ caminho 2 = 0,8666.

Logo, os caminhos ficam ordenados da seguinte forma: caminho 1, caminho 2.

Vale ressaltar que, caso fossem obtidos mais de dois caminhos entre um par de nós, a análise dos resultados para a ordenação dos caminhos não-dominados seria análoga, sendo prevista no Passo 5 do algoritmo.

Para fins de ilustração e comparação, este problema incerto foi transformado em um problema *crisp*, ou seja, foram considerados os valores modais para os custos e as capacidades, obtendo os seguintes resultados:

- Caminho 1: $1 \rightarrow 2 \rightarrow 5$, com custo = 51, tempo = 4.
- Caminho 2: $1 \rightarrow 2 \rightarrow 4 \rightarrow 5$, com custo = 50, tempo = 12 (eliminado, tempo > 11).

Analisando os caminhos obtidos verificou-se que o segundo foi eliminado por não satisfazer as restrições de tempo dos nós, ou seja, seu tempo foi maior que o limite exigido.

Com os resultados obtidos pelos dois problemas (*fuzzy* e *crisp*) percebe-se que os do primeiro (problema *fuzzy*) são mais completos que os do segundo (problema *crisp*), pois, caso o usuário queira optar pelo custo, não levando em consideração o tempo, escolherá o Caminho 2. Porém, caso seja considerado somente o problema *crisp*, o usuário deverá, necessariamente, optar pelo Caminho 1, mesmo tendo um custo maior.

Exemplo 2.

Este exemplo (Figura 5) se refere à rede Europeia COST239 (Tan & Sinclair, 1995). Os nós da rede e suas restrições de tempo estão definidos na Tabela 2.

Tabela 2 – Restrição de tempo dos nós da rede COST239.

Nós	Cidades	Restrições – Tempo
1	Paris	(0 0 0)
2	Milão	(1,6 1,7 2,0)
3	Zurique	(0,9 1,0 1,3)
4	Praga	(1,4 1,7 1,9)
5	Viena	(2,0 2,2 2,5)
6	Berlim	(0,8 1,0 1,1)
7	Amsterdã	(0,7 0,8 0,9)
8	Luxemburgo	(0,9 1,0 1,1)
9	Bruxelas	(0,3 0,45 0,5)
10	Londres	(1,0 1,2 1,3)
11	Copenhague	(1,65 1,95 2,0)

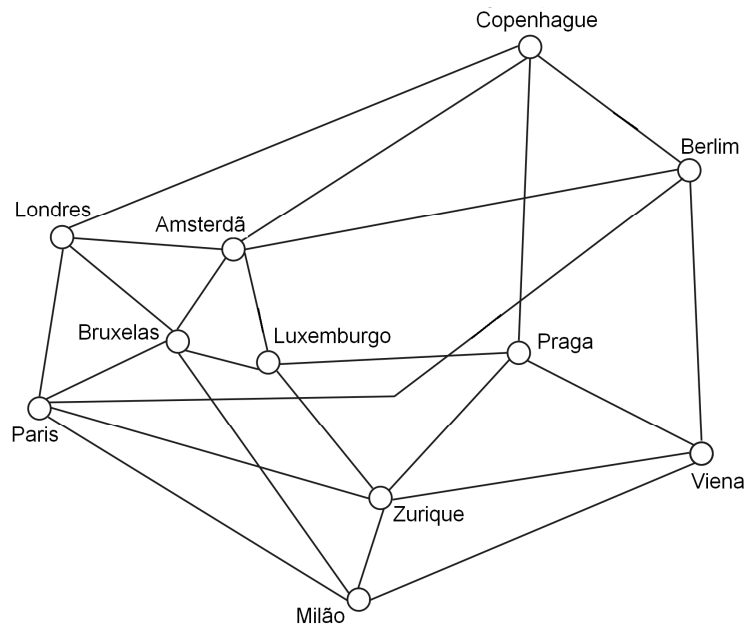


Figura 5 – Rede Europeia – COST239.

Na Tabela 3 estão os nós origem e destino de cada arco, bem como os seus respectivos custos e tempos, enquanto na Tabela 4 são apresentados os resultados finais.

Tabela 3 – Dados da rede COST239.

Arco	Origem	Destino	Custos	Tempos
1	1	2	(800 820 840)	(1,5 1,7 1,9)
2	1	3	(350 361 370)	(0,8 0,9 1,0)
3	1	6	(650 677 683)	(0,95 1,05 1,15)
4	1	9	(290 300 350)	(0,4 0,52 0,6)
5	1	10	(420 450 470)	(0,95 1,0 1,3)
6	2	3	(180 186 193)	(0,35 0,45 0,5)
7	2	5	(495 510 525)	(1,25 1,5 1,9)
8	2	9	(900 930 960)	(1,7 2,0 2,1)
9	3	4	(650 667 863)	(0,9 0,98 1,2)
10	3	5	(730 748 770)	(1,15 1,3 1,45)
11	3	8	(425 443 465)	(0,5 0,6 0,7)
12	4	5	(190 199 210)	(0,2 0,3 0,4)
13	4	6	(310 340 360)	(0,5 0,65 0,7)
14	4	11	(710 740 770)	(1,05 1,15 1,2)
15	5	6	(610 660 690)	(0,75 0,85 0,96)
16	6	11	(230 242 260)	(0,8 0,9 1,0)
17	7	6	(390 410 440)	(0,75 0,85 0,96)
18	7	11	(450 472 490)	(1,0 1,16 1,3)
19	8	4	(710 730 735)	(1,05 1,2 1,25)
20	8	7	(230 242 255)	(0,3 0,42 0,53)
21	9	8	(130 137 145)	(0,23 0,3 0,35)
22	9	7	(120 130 150)	(0,25 0,28 0,32)
23	9	10	(230 242 260)	(0,3 0,42 0,53)
24	10	7	(330 342 350)	(0,5 0,65 0,7)
25	10	11	(1250 1310 1430)	(2,30 2,45 2,6)

Tabela 4 – Resultados da rede COST239.

Origem	Destino	Caminho	Custo Caminho	Tempos	Pc	Pt
1	2	1→2	(800 820 840)	(1,5 1,7 1,9)	-	1,0
1	3	1→3	(350 361 370)	(0,8 0,9 1,0)	-	1,0
1	4	1→3→4	(1000 1028 1233)	(1,7 1,88 2,2)	1,0	0,5263
1	4	1→9→8→4	(1130 1167 1230)	(1,7 2,02 2,2)	0,333	0,3846
1	5	1→3→5	(1080 1109 1140)	(1,95 2,2 2,45)	-	1,0
1	6	1→6	(650 677 683)	(0,95 1,05 1,15)	-	0,75
1	7	1→9→7	(410 430 500)	(0,65 0,8 0,92)	-	1,0
1	8	1→9→8	(420 437 495)	(0,65 0,82 0,95)	-	1,0
1	9	1→9	(290 300 350)	(0,4 0,52 0,6)	-	0,59
1	10	1→10	(420 450 470)	(0,95 1,0 1,3)	-	1,0
1	11	1→6→11	(880 919 943)	(1,75 1,95 2,15)	0,893	1,0
1	11	1→9→7→11	(860 902 990)	(1,65 1,96 2,22)	1,0	0,9722

Analisando a Tabela 4 verifica-se que existem dois caminhos não-dominados entre os nós 1 e 4 ($1 \rightarrow 3 \rightarrow 4$; $1 \rightarrow 9 \rightarrow 8 \rightarrow 4$) e entre os nós 1 e 11 ($1 \rightarrow 6 \rightarrow 11$; $1 \rightarrow 9 \rightarrow 7 \rightarrow 11$) que satisfazem as restrições de tempo. Para os demais nós existe apenas um caminho não-dominado.

Verificando os dois caminhos entre os nós 1 e 4 percebe-se que o primeiro ($1 \rightarrow 3 \rightarrow 4$) é melhor do que o segundo, visto que tanto sua medida de possibilidade em relação ao custo (P_c), quanto em relação ao tempo (P_t), são maiores do que as medidas do segundo caminho ($1 \rightarrow 9 \rightarrow 8 \rightarrow 4$).

A análise dos caminhos entre os nós 1 e 11 é semelhante à feita em relação aos caminhos entre os nós 1 e 4, porém ficará a cargo do usuário decidir se deve priorizar o tempo ou o custo, semelhante ao Exemplo 1.

Ressalta-se que não foram calculadas as possibilidades de custo dos demais caminhos em virtude da existência de um único caminho entre os respectivos nós.

Assim, como o exemplo anterior, este foi convertido em um problema clássico, sendo os custos dos arcos e os tempos os valores modais. Após executá-lo, foram obtidos os seguintes resultados:

Tabela 5 – Resultados do problema clássico.

Origem	Destino	Caminho	Custo Caminho	Tempo
1	2	$1 \rightarrow 2$	820	1,7
1	3	$1 \rightarrow 3$	361	0,9
1	4	-	-	-
1	5	$1 \rightarrow 3 \rightarrow 5$	1109	2,2
1	6	-	-	-
1	7	$1 \rightarrow 9 \rightarrow 7$	430	0,8
1	8	$1 \rightarrow 9 \rightarrow 8$	437	0,82
1	9	-	-	-
1	10	$1 \rightarrow 10$	450	1,0
1	11	$1 \rightarrow 6 \rightarrow 11$	919	1,95

Verificando os resultados da Tabela 5 percebe-se que não foi possível encontrar caminhos entre todos os pares de nós [(1,4), (1,6) e (1,9)], pois os caminhos existentes não satisfizeram as restrições de tempo. Este fato pode ser comprovado observando a Tabela 4, haja vista que todos os caminhos existentes entre estes pares de nós possuem $P_t < 1$.

Outro fato importante pode ser observado no caminho encontrado entre os nós 1 e 11: não foi escolhido o que possui o menor custo, pois o mesmo não satisfez as restrições de tempo.

Comparando os resultados dos dois problemas, *fuzzy* e clássico, verificou-se que o primeiro apresentou um gama de resultados mais significativa.

5. Conclusões

Em problemas de grafos com incertezas, o problema de caminho mínimo *fuzzy* é um dos mais estudados, visto que apresenta aplicações nas mais diversas áreas, fato este que pode ser comprovado pelas várias citações do texto.

Neste trabalho foi proposto um algoritmo para o PCM-*fuzzy* com dois parâmetros incertos nos arcos, custo e tempo, além da restrição de tempo dos nós. O algoritmo proposto apresenta como solução final um conjunto ordenado de caminhos não-dominados, ao contrário dos trabalhos da literatura que, quando abordam o tema desta forma (utilizando caminhos não-dominados), apenas apresentam o conjunto solução, sem sequer guiar o usuário para a escolha de um dos caminhos.

Vale ressaltar que o único trabalho da literatura que aborda o PCM-*fuzzy* com restrição de tempo (Hernandes & Takahashi, 2004) não apresenta o conjunto solução ordenado além de possuir uma complexidade computacional maior. É uma adaptação do algoritmo de Ford-Moore-Bellman, enquanto este é uma adaptação do algoritmo de Dijkstra.

Como trabalho futuro, pretende-se adaptar este problema a redes que possuam parâmetros negativos nos arcos e aplicar α -cortes no conjunto solução dos caminhos não-dominados. Outra proposta é aplicar este algoritmo no Problema de Fluxo de Custo Mínimo *Fuzzy*.

Referências Bibliográficas

- (1) Bassanezi, R.C. & Barros, L.C. (2006). *Tópicos de Lógica Fuzzy e Biomatemática*. Unicamp / IMECC, Campinas, SP.
- (2) Bellman, R.E. (1958). On a routing problem. *Quarterly Applied Mathematics*, **16**, 87-90.
- (3) Bellman, R.E. & Zadeh, L. (1970). Decision making in a fuzzy environment. *Manag. Science*, **17**.
- (4) Bezdek, J. (1981). *Pattern Recognition with Fuzzy Objective Function Algorithms*. Plenum Press.
- (5) Blue, M.; Bush, B. & Puckett, J. (2002). Unified approach to fuzzy graph problems. *Fuzzy Sets and Systems*, **125**, 355-368.
- (6) Chanas, S.; Delgado, M.; Verdegay, J.L. & Vila, M. (1995). Fuzzy optimal flow on imprecise structures. *European Journal of Operational Research*, **83**, 568-580.
- (7) Chuang, T. & Kung, J. (2005). The fuzzy shortest path length and the corresponding shortest path in a network. *Computers and Operations Research*, **32**, 1409-1428.
- (8) Dubois, D. & Prade, H. (1980). *Fuzzy Sets and Systems: Theory and Applications*. Academic Press, New York.
- (9) Eppstein, D. (1994). Finding the k-shortest paths. *Anais do IEEE Symposium on Foundations of Computer Science*, 154-165.
- (10) Goldberg, M.C. & Luna, H.P.L. (2000). *Otimização Combinatória e Programação Linear – Modelos e Algoritmos*. Editora Campus Ltda., Rio de Janeiro, RJ.

- (11) Hansen, P. (1980). Bicriterion path problems. **In:** *Multiple criteria decision making: theory and applications, lecture note in economics and in mathematical systems*. [edited by M. Beckmann and H.P. Kunzi], Springer Verlag, **177**, 109-127.
- (12) Hernandez, F. & Takahashi, M.T. (2004). Adaptação de um algoritmo de caminho mínimo em grafos com custo nos arcos e restrições de tempo *fuzzy*. *Anais do XXXVI Simpósio Brasileiro de Pesquisa Operacional*.
- (13) Hernandez, F.; Yamakami, A.; Takahashi, M.T. & Verdegay, J.L. (2006). Um algoritmo para o problema de fluxo máximo em redes com incertezas. *Anais do XXXVIII Simpósio Brasileiro de Pesquisa Operacional*, 2272-2283.
- (14) Hernandez, F. (2007). Algoritmos para grafos com incertezas. Tese de doutorado. Faculdade de Engenharia Elétrica e de Computação, Universidade Estadual de Campinas.
- (15) Hernandez, F.; Lamata, M.T.; Verdegay, J.L. & Yamakami, A. (2007). The shortest problem on networks with fuzzy parameters. *Fuzzy Sets and Systems*, **158**, 1561-1570.
- (16) Kim, K. & Roush, F. (1981). Fuzzy flows on networks. *Fuzzy Sets and Systems*, **8**, 35-38.
- (17) Klein, C.M. (1991). Fuzzy shortest paths. *Fuzzy Sets and Systems*, **39**, 27-41.
- (18) Lin, C. & Chern, M.S. (1993). The fuzzy shortest path problem and its most vital arcs. *Fuzzy Sets and Systems*, **58**, 343-353.
- (19) Malik, D. & Moderson, J.N. (2001). *Fuzzy Discrete Structures*. Volume 58 da série de livros Studies in Fuzziness and Soft Computing, Springer-Verlag, New-York.
- (20) Moderson, J.N. & Nair, P.S. (2000). *Fuzzy graphs and fuzzy hypergraphs*. Livro da série: Studies in Fuzziness and Soft Computing, Physica-Verlag.
- (21) Nayeem, S.M.A. & Pal, M. (2005). Shortest path problem on a network with imprecise edge weight. *Fuzzy Optimization and Decision Making*, **4**, 293-312.
- (22) Okada, S. & Soper, T. (2000). A shortest path problem on a network with fuzzy arc lengths. *Fuzzy Sets and Systems*, **109**, 129-140.
- (23) Okada, S. (2004). Fuzzy shortest path problems incorporating interactivity among paths. *Fuzzy Sets and Systems*, **142**(3), 335-357.
- (24) Pedrycz, W. & Gomide, F. (1998). *An Introduction to Fuzzy Sets: Analysis and Design*. MIT Press, London.
- (25) Sengupta, A. & Pal, T.K. (2000). On comparing interval numbers. *European Journal of Operational Research*, **127**, 29-43.
- (26) Shaw, I.S. & Simões, M.G. (1998). *Controle e Modelagem Fuzzy*. Edgar-Blucher/Fapesp.
- (27) Shih, H.-S. & Lee, E.S. (1999). Fuzzy multi-level minimum cost flow problems. *Fuzzy Sets and Systems*, **107**, 159-176.
- (28) Takahashi, M.T. (2004). Contribuições ao estudo de grafos fuzzy: teoria e algoritmos. Tese de doutorado, Faculdade de Engenharia Elétrica e de Computação, Universidade Estadual de Campinas.

- (29) Takahashi, M.T. & Yamakami, A. (2004). The assignment problem with fuzzy weights. *Anais do North American Fuzzy Information Processing Society*, Alberta, Canadá.
- (30) Tan, L.G. & Sinclair, M.C. (1995). Wavelength assignment between the central nodes of the cost239 european optical network. *Anais do 11th UK Performance Engineering Workshop*, Liverpool, 235-247.
- (31) Zadeh, L. (1965). Fuzzy sets. *Information and Control*, **8**, 338-353.
- (32) Zadeh, L. (1968). Fuzzy algorithms. *Information and Control*, **12**.
- (33) Zadeh, L. (1978). Fuzzy sets as a theory of possibility. *Journal of Fuzzy Sets and Systems*, **1**.