

## Problema do Caminho Mínimo Fuzzy

---

Este é um dos problemas mais antigo e mais *importante* da teoria de grafos/fluxos em redes pois existe uma grande variedade de aplicações (telecomunicações, transporte, etc.), possui algoritmos eficientes (por exemplo, Dijkstra, Floyd, Ford) e pode ser usado como base para estudos mais complexos.

Seja um grafo  $G : (N, A)$  com  $m$  nós e  $n$  arcos com custo  $c_{ij}$  associado aos arcos.

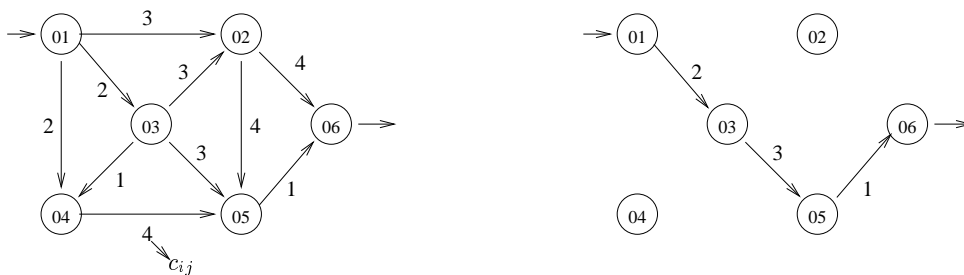


Figura 1: Exemplo de um problema de caminho mínimo

O *comprimento* de um caminho  $P$  do grafo é dado por:  $l_P = \sum_{(i,j) \in P} c_{ij}$ .

## *Definição do caso clássico*

---

O problema do caminho mínimo pode consistir em:

- encontrar o caminho de menor comprimento no grafo entre o nó 1 e o nó  $m$
- Encontrar um caminho mínimo entre um nó e todos os outros nós.
- Encontrar um caminho mínimo entre todos os nós.

As seguintes suposições são feitas

1. O grafo  $G$  possui um caminho de um nó a todos os outros nós.
2. O grafo não possui ciclos negativos.
3. O grafo é direcionado.

## *Problema de caminho mínimo fuzzy: dois casos*

---

Um problema pode ter diferente níveis de fuzzissidade. Existem dois problemas principais que devem ser abordados e são citados por todos os trabalhos encontrados na literatura:

1. A estrutura do grafo é bem definida (grafo crisp) e os parâmetros associados são números fuzzy. É o mais estudado.
2. O comprimento do caminho é um número fuzzy e cada arco do grafo tem um valor de pertinência, isto é, a cada nó e arco existe um conjunto fuzzy associado.

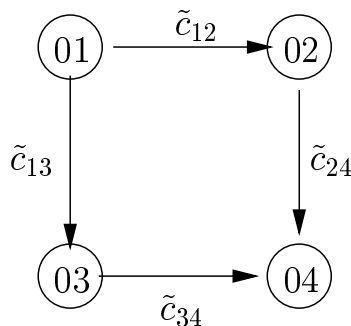
Serão apresentados alguns algoritmos para o problema de caminho mínimo fuzzy (grafo crisp e parâmetros fuzzy).

## *Dubois e Prade, 1980*

---

- Uma das primeiras referências que tratam do problema de caminho mínimo fuzzy
- Propõe a simples extensão de dois algoritmos clássicos (Floyd e de Ford)
- Ponto fraco: caminho encontrado pelos algoritmos pode não existir.

**Exemplo:** Seja uma rede fuzzy tendo o nó 1 como origem e o nó 4 como destino.



$$\begin{aligned}c_{12} &= \left\{ \frac{2}{0.8}, \frac{3}{0.3} \right\}, \\c_{13} &= \left\{ \frac{1}{0.5}, \frac{2}{0.6}, \frac{3}{0.2} \right\}, \\c_{24} &= \left\{ \frac{1}{0.2}, \frac{2}{0.3} \right\} \text{ e} \\c_{34} &= \left\{ \frac{1}{0.2}, \frac{2}{0.8} \right\}.\end{aligned}$$

### **Caminhos possíveis:**

(1 → 2 → 4) com comprimento {3/0.2, 4/0.3, 5/0.3}

(1 → 3 → 4) com comprimento {2/0.2, 3/0.5, 4/0.6, 5/0.2}

## *Dubois e Prade, 1980 (cont.)*

---

**Algoritmo:**

**Passo 0:**  $f(m) \leftarrow \{0/1\}$

**Passo 1:** Para o nó  $m - 1$  ao nó 1, faça

$$f(i) \leftarrow \widetilde{\min}_{k:k>i} \{ \widetilde{c}_{ik} \widetilde{+} f(k) \}$$

sendo que o valor associado ao nó 1 é a solução do problema.

Usando o algoritmo acima, temos:

$$\widetilde{f}(4) = \{0/1\},$$

$$\begin{aligned} \widetilde{f}(3) &= \{ \widetilde{c}_{34} + \widetilde{f}(4) \} = \{1/0.2, 2/0.8\} + \{0/1\}, \\ &= \{1/0.2, 2/0.8\}. \end{aligned}$$

$$\widetilde{f}(2) = \{ \widetilde{c}_{24} + \widetilde{f}(4) \} = \{1/0.2, 2/0.3\}$$

$$\begin{aligned} \widetilde{f}(1) &= \widetilde{\min} \{ \widetilde{c}_{12} + \widetilde{f}(2), \widetilde{c}_{13} + \widetilde{f}(3) \} \\ &= \{3/0.2, 4/0.3, 5/0.3\} \widetilde{\wedge} \{2/0.2, 3/0.5, 4/0.6, 5/0.2\} \\ &= \{2/0.2, 3/0.3, 4/0.3, 5/0.2\} \end{aligned}$$

- Não existe caminho de comprimento  $\{2/0.2, 3/0.3, 4/0.3, 5/0.2\}$  do nó 1 ao nó 4.
- Note também que nenhum caminho tem valor de pertinência 0.3 para o comprimento 3  
 $\rightarrow \{3/0.2, 4/0.3, 5/0.3\} \widetilde{\wedge} \{3/0.5\} = \vee((0.2 \wedge 0.5, 0.3 \wedge 0.5, 0.3 \wedge 0.5))$

- Contorna a dificuldade verificada por Dubois e Prade com o uso de dominância de conjuntos fuzzy (tipo *more is better*).
- considera conjuntos discretos fuzzy para os comprimentos dos arcos
- determina-se o comprimento máximo que pode alcançar um caminho e trabalha-se com este valor no algoritmo de programação dinâmica

$$\left. \begin{array}{l} M : \text{ número de camadas} = 3 \\ R : \text{ comprimento máximo dos arcos} = 3 \end{array} \right\} (M - 1) \cdot R = 6$$

**Algoritmo:**

**Passo 0:**  $f(m) \leftarrow (1, \dots, 1)$  (vetor com  $(M - 1)R$  posições)

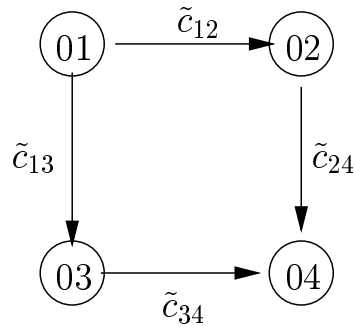
**Passo 1:** Para  $i = (m - 1)$  até 1, faça

$$f(i) \leftarrow \text{dom}_{k:(i,k) \in A} (\tilde{c}_{ik} \tilde{+} f(k))$$

onde o operador de dominância deve escolher o vetor com os maiores valores de pertinência.

*Klein, 1991 (cont.)*

---



$$\begin{aligned}
 c_{12} &= \left\{ \frac{2}{0.8}, \frac{3}{0.3} \right\}, \\
 c_{13} &= \left\{ \frac{1}{0.5}, \frac{2}{0.6}, \frac{3}{0.2} \right\}, \\
 c_{24} &= \left\{ \frac{1}{0.2}, \frac{2}{0.3} \right\} \text{ e} \\
 c_{34} &= \left\{ \frac{1}{0.2}, \frac{2}{0.8} \right\}.
 \end{aligned}$$

Para o exemplo anterior, temos:

$$f(4) = (1, 1, 1, 1, 1, 1)$$

$$f(3) = (0.2, 0.8, 0, 0, 0, 0)$$

$$f(2) = (0.2, 0.3, 0, 0, 0, 0)$$

$$f(1) = \text{dom}\{\tilde{c}_{13} \tilde{+} f(3), \tilde{c}_{12} \tilde{+} f(2)\}$$

$$= \text{dom}\{(0, 0.2, 0.5, 0.6, 0.2, 0), (0, 0, 0.2, 0.3, 0.3, 0)\}$$

$$= (0, 0.2, 0.5, 0.6, 0.2, 0)$$

**Caminho escolhido:**  $(1 \rightarrow 3 \rightarrow 4)$  com comprimento  $\left\{ \frac{2}{0.2}, \frac{3}{0.5}, \frac{4}{0.6}, \frac{5}{0.2} \right\}$ .

- considera os comprimentos dos arcos como números fuzzy
- utiliza teoria de possibilidade
- resolve um problema multiobjetivo
- encontra um conjunto de caminhos Pareto-ótimos (não-dominados)

**Informações para o algoritmo:**

**T:** conjunto temporário de nós onde cada elemento  $(i, k)$  é a  $k$ -ésima etiqueta do nó  $i$

**P:** conjunto permanente de nós

$[\tilde{d}_l(p_{sj}), (i, k)]_l$ : é a  $l$ -ésima etiqueta do nó  $j$ .  $\tilde{d}_l(p_{sj})$  é a distância do nó  $s$  ao nó  $j$  e  $(i, k)$  apresenta o nó precedente com dados apresentados na etiqueta  $k$  deste nó.

$\tilde{d}_l(p_{sj}) \prec \tilde{d}_u(p_{sj})$ : o comprimento do caminho do nó  $s$  ao nó  $j$  dado pela etiqueta  $l$  domina o comprimento dado pela etiqueta  $u$ .



## *Algoritmo de Okada e Soper, 2000 (cont.)*

---

**Passo 0:** Associe a etiqueta  $[\tilde{0}, (-, -)]$  para o nó  $s$ . Faça

$$T \leftarrow (1, 1) \text{ e } P \leftarrow \emptyset$$

**Passo 1:** (Seleção das etiquetas) Se  $T = \emptyset$  FIM. Caso contrário, entre as etiquetas temporárias, determine qual é a lexicograficamente menor. Seja o  $k$ -ésima etiqueta associada ao nó  $i$ . Faça

$$T \leftarrow T - \{(i, k)\} \text{ e } P \leftarrow P \cup (i, k)$$

**Passo 2:** (Verifica a etiqueta e se existe dominância) Para cada nó  $j \in N$  tal que  $(i, j) \in A$ , faça

(i) Determine a distância do caminho fuzzy

$$\tilde{d}_l(p_{sj}) = \tilde{d}_k(p_{si}) \oplus \tilde{l}_{ij}$$

(ii) Seja  $[\tilde{d}_l(p_{sj}), (i, k)]_l$  uma nova etiqueta temporária do nó  $j$ . Atualize o conjunto  $T$

$$T \leftarrow T \cup (j, l)$$

(iii) Para cada  $\tilde{d}_u(p_{sj})$  tal que  $(j, u) \in T \cup P$ . Se  $\tilde{d}_l(p_{sj}) \succ \tilde{d}_u(p_{sj})$  então elimine a  $l$ -ésima etiqueta sobre o nó  $j$ , retire  $(j, l)$  do conjunto  $T$ .

Para cada  $\tilde{d}_u(p_{sj})$  tal que  $(j, u) \in T$ . Se  $\tilde{d}_l(p_{sj}) \prec \tilde{d}_u(p_{sj})$ , então elimine a  $u$ -ésima etiqueta do nó  $j$  e retire  $(j, u)$  do conjunto  $T$ .

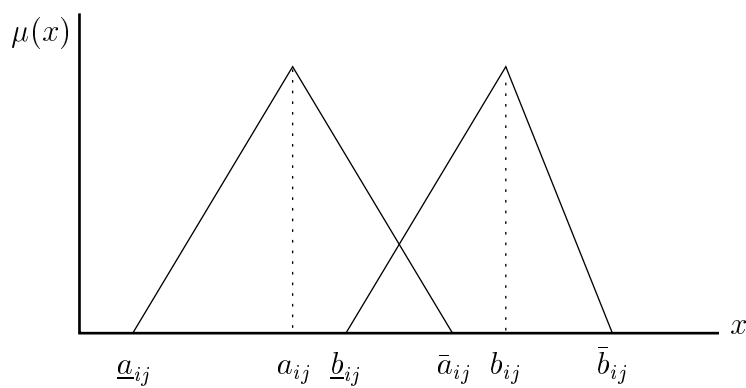
**Passo 3:** Volte ao Passo 1.

## Okada e Soper, 2000 (cont.)

---

Um valor  $\tilde{a}_{ij} = (\underline{a}_{ij}, a_{ij}, \bar{a}_{ij})$  é lexicograficamente menor do que  $\tilde{b}_{ij} = (\underline{b}_{ij}, b_{ij}, \bar{b}_{ij})$  se uma das regras abaixo valer:

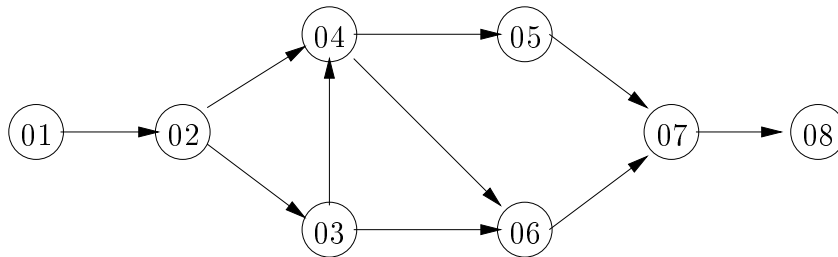
- (i)  $\underline{a}_{ij} < \underline{b}_{ij}$
- (ii)  $\underline{a}_{ij} = \underline{b}_{ij}$  e  $a_{ij} < b_{ij}$
- (iii)  $\underline{a}_{ij} = \underline{b}_{ij}$ ,  $a_{ij} = b_{ij}$  e  $\bar{a}_{ij} \leq \bar{b}_{ij}$



## Okada e Soper, 2000 (cont.)

---

Exemplo:



arco	comprimento	arco	comprimento
(1,2)	(19,25,29)	(3,6)	(54,57,62)
(2,3)	(15,20,25)	(4,6)	(8,9,10)
(2,4)	(58,63,68)	(5,7)	(70,75,80)
(3,4)	(38,41,46)	(6,7)	(65,75,85)
(4,5)	(12,15,18)	(7,8)	(20,25,30)

Tabela 1: Exemplo de um grafo com parâmetros fuzzy  $(\underline{a}_{ij}, a_{ij}, \bar{a}_{ij})$

## *Okada e Soper, 2000 (cont.)*

---

**Passo 0:**

$$[01] \leftarrow [(0, 0, 0), (-, -)]_1$$

$$T \leftarrow (1, 1) \text{ e } P \leftarrow \emptyset$$

**Passos 01 e 02:**

nó [01] é lexicograficamente menor

$$T \leftarrow \emptyset, P \leftarrow (1, 1)$$

$$\tilde{d}_1(p_{12}) = (0, 0, 0) \oplus (19, 25, 29) = (19, 25, 29)$$

$$T \leftarrow T \cup (2, 1), [02] \leftarrow [(19, 25, 29), (1, 1)]_1$$

nó [02] é lexicograficamente menor

$$T \leftarrow \emptyset, P \leftarrow \{(1, 1), (2, 1)\}$$

$$\tilde{d}_1(p_{13}) = (19, 25, 29) \oplus (15, 20, 25) = (34, 45, 54)$$

$$T \leftarrow T \cup (3, 1), [03] \leftarrow [(34, 45, 54), (2, 1)]_1$$

$$\tilde{d}_1(p_{14}) = (19, 25, 29) \oplus (58, 63, 68) = (77, 88, 97)$$

$$T \leftarrow T \cup (4, 1), [04] \leftarrow [(77, 88, 97), (2, 1)]_1$$

ainda não existe comparações de caminhos dominados.

## *Algoritmo de Okada e Soper, 2000 (cont.)*

---

nó [03] é lexicograficamente menor

$$T \leftarrow \{(4, 1)\}, P \leftarrow \{(1, 1), (2, 1), (3, 1)\}$$

$$\tilde{d}_2(p_{14}) = (34, 45, 54) \oplus (38, 41, 46) = (62, 86, 100)$$

$$T \leftarrow T \cup (4, 2), [04] \leftarrow [(62, 86, 100), (3, 1)]_2$$

$$\tilde{d}_1(p_{16}) = (34, 45, 54) \oplus (54, 57, 62) = (88, 102, 116)$$

$$T \leftarrow T \cup (6, 1), [06] \leftarrow [(88, 102, 116), (6, 1)]_1$$

não existem caminhos dominados.

nó [04] é lexicograficamente menor

$$T \leftarrow \{(4, 1), (6, 1)\}, P \leftarrow \{(1, 1), (2, 1), (3, 1), (4, 2)\}$$

$$\tilde{d}_1(p_{15}) = (62, 86, 100) \oplus (12, 15, 18) = (74, 101, 118)$$

$$T \leftarrow T \cup (5, 1), [05] \leftarrow [(74, 101, 118), (4, 2)]_1$$

$$\tilde{d}_2(p_{16}) = (62, 86, 100) \oplus (8, 9, 10) = (70, 95, 110)$$

$$T \leftarrow T \cup (6, 2), [06] \leftarrow [(70, 95, 110), (4, 2)]_2$$

Neste caso  $\tilde{d}_2(p_{16}) \prec \tilde{d}_1(p_{16})$ .

$$T \leftarrow T \setminus (6, 1) = \{(4, 1), (5, 1), (6, 2)\}$$

## *Okada e Soper, 2000 (cont.)*

---

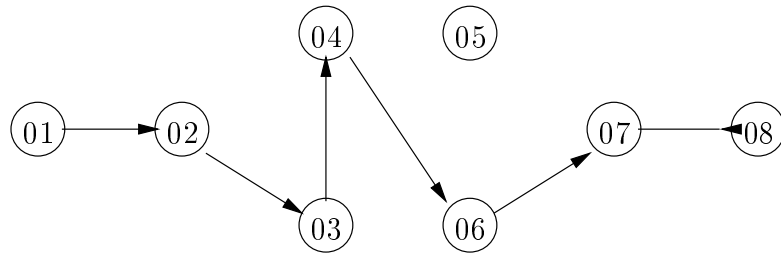
nó	etiqueta	comprimento	precedente
01	1	(0,0,0)	(-, -)
02	1	(19,25,29)	(1,1)
03	1	(34,45,54)	(2,1)
04	1	(77,88,97)	(2,1)
04	2	(62,86,100)	(3,1)
06*	1	(88,102,116)	(3,1)
05	1	(74,101,118)	(4,2)
06	2	(70,95,110)	(4,2)
07	1	(135,170,195)	(6,2)
07*	2	(144,176,198)	(5,1)
05	2	(89,103,115)	(4,1)
06	3	(85,97,107)	(4,1)
07	3	(150,172,192)	(6,3)
07*	4	(164,178,195)	(5,2)
08	1	(155,195,225)	(7,1)
08	2	(170,197,222)	(7,3)

Tabela 2: Resultados do algoritmo (\* elementos eliminados)

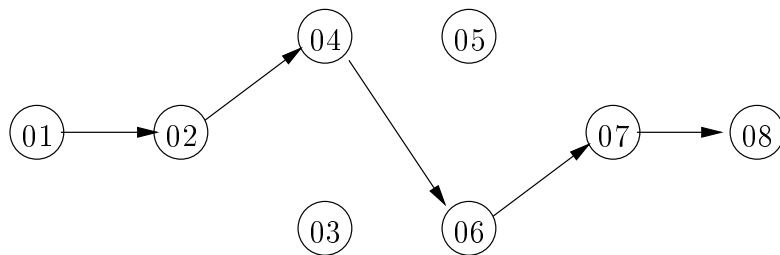
*Okada e Soper, 2000 (cont.)*

---

Soluções para o problema:



Comprimento = (155,195,225)



Comprimento = (170,197,222)

- Considera os comprimentos dos arcos como números fuzzy.
- Também utiliza teoria de possibilidade.
- Com base em algumas propriedades do problema, introduz o conceito de **grau de possibilidade de um arco pertencer ao caminho mínimo** de um nó origem  $s$  a um nó destino  $t$ .
- O método de Dijkstra generalizado é usado para encontrar os caminhos.



## *Algoritmo proposto por Okada, 2001*

---

Seja  $L_r$  a lista de etiquetas  $[i, k]$  ( $i$  é o nó que precede  $r$  e  $k$ ,  $k$ -ésima etiqueta na lista  $L_i$ ). Esta lista é usada para compor o caminho de 1 a  $r$ . Definimos também  $\mathcal{P}_{1r}^\alpha$  como o conjunto de caminhos compostos da lista  $L_r$  com respeito ao limitante  $\alpha$ . Seja também  $B_r$  o conjunto de nós que precedem  $r$ . O Algoritmo proposto consiste de dois procedimentos. No primeiro, o grau de possibilidade aproximado para todo o arco é determinado pela repetição do procedimento 2, descrito a seguir, para cada  $\alpha \in [0, 1]$ . No segundo procedimento, os arcos cujo grau de possibilidade são maiores que um  $\alpha$  dado, são identificados.

## *Okada, 2001: Procedimento 01*

---

### **Procedimento 01:**

**Passo 0:** Defina um número de partições  $h \in \mathbb{Z}^+$  dividindo o intervalo  $[0,1]$ .

$$D_{ij} \leftarrow 0, \forall (i, j) \in A$$

$$A' \leftarrow A$$

**Passo 1:** Para o parâmetro  $\alpha = \frac{1}{h}, \frac{2}{h}, \dots, 1$ , repita os passos 2 e 3.

**Passo 2:** Execute o Procedimento 2 com o parâmetro  $\alpha$ .

**Passo 3:**  $A' \leftarrow A' - \{(i, j) | (i, j) \in A', D_{ij} < \alpha\}$ '

**Passo 4:** Apresente  $D_{ij}$  para todo  $(i, j) \in A$ . FIM.

## Okada, 2001: Procedimento 02

---

### Procedimento 2:

**Passo 0:** Compute  $(\tilde{l}_{ij})_\alpha$  ( $\forall (i, j) \in A'$ ).

**Passo 1:**  $L_1 \leftarrow [-, -]$ .  $L_j \leftarrow \emptyset$ , ( $j = 2, \dots, n$ ).

**Passo 2:** Para  $r = 2, \dots, n$ , repita os passos 3, 4 e 5.

**Passo 3:** Para  $j \in B_r$ ,  $L'_j \leftarrow L_j$  e  $i$  em todo  $[i, k] \in L'_j$  é substituído por  $j$ .

$$L_r \leftarrow \bigcup_{j \in B_r} L'_j$$

**Passo 4:** Se  $|B_r| = 1$  ( $r$  tem somente um nó precedente), então retorne ao passo 2. Caso contrário vá ao passo 5.

**Passo 5:** Para cada par de etiquetas  $[i_s, k_s], [i_t, k_t] \in L_r$ , componha os caminhos  $p_s$  e  $p_t$  respectivamente voltando dos nós  $s$  e  $t$  até o nó 1.

Se  $Poss(\sum_{(i,j) \in p_s} l_{ij} \leq \sum_{(i,j) \in p_t} l_{ij}) < \alpha$ , então  $L_r \leftarrow L_r - [i_s, k_s]$ .

**Passo 6:**  $D_{ij} \leftarrow \alpha$ ,  $\forall (i, j) \in p$ ,  $p \in \mathcal{P}_{1n}^\alpha$ . Volte ao Procedimento 1.

## Okada, 2001: Exemplo

---

O exemplo será o mesmo resolvido para o algoritmo de Okada e Soper.

**(Procedimento 01) Passo 0:**  $h = 10$ ,  $D_{ij} = 0$ ,  $\forall (i, j) \in A$  e  $A' \leftarrow A$

**Passo 1:**  $\alpha \leftarrow 0, 1$ . Entrar no Procedimento 02.

$$\mathbf{r=2} \quad B_2 = \{1\}, L_2 \leftarrow \{[1, 1]\}$$

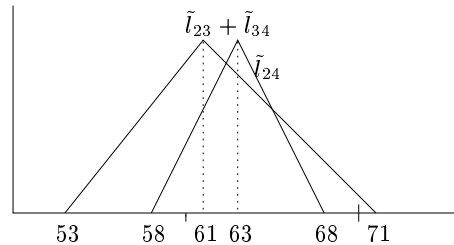
$$\mathbf{r=3} \quad B_3 = \{2\}, L_3 \leftarrow \{[2, 1]\}$$

$$\mathbf{r=4} \quad B_4 = \{2, 3\}, L_4 \leftarrow \{[2, 1], [3, 1]\}$$

$$1 \rightarrow 2 \rightarrow 4 \text{ com } \tilde{l} = (77, 88, 97)$$

$$1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \text{ com } \tilde{l} = (72, 86, 100)$$

$$Poss(\sum_{(i,j) \in P_1} l_{ij} \leq \sum_{(i,j) \in P_2} l_{ij}) = Poss(l_{24} \leq l_{23} + l_{34}) > 0, 1$$



$$\mathbf{r=5} \quad B_5 = \{4\}, L_5 \leftarrow \{[4, 1]\}$$

$$\mathbf{r=6} \quad B_6 = \{3, 4\}, L_4 \leftarrow \{[2, 1], [4, 1]\}$$

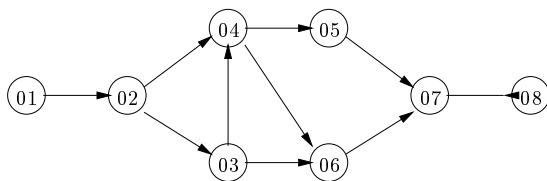
$$Poss(\sum_{(i,j) \in P_1} l_{ij} \leq \sum_{(i,j) \in P_2} l_{ij}) = Poss(l_{23} + l_{36} \leq l_{24} + l_{46}) > 0, 1$$

$$Poss(\sum_{(i,j) \in P_1} l_{ij} \leq \sum_{(i,j) \in P_3} l_{ij}) = Poss(l_{36} \leq l_{34} + l_{46}) = 0, 22$$

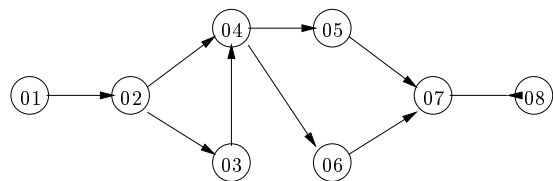
$$Poss(\sum_{(i,j) \in P_2} l_{ij} \leq \sum_{(i,j) \in P_3} l_{ij}) = Poss(l_{24} \leq l_{23} + l_{34}) > 0, 1$$

## Okada, 2001: Exemplo

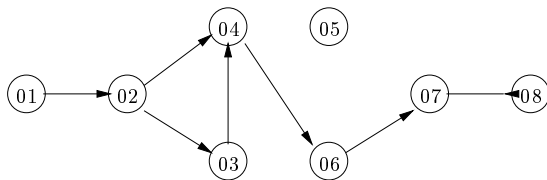
continuando algoritmo temos que para  $\alpha = 0, 1$  e  $\alpha = 0, 2$  os valores de  $D_{ij} = \alpha$ . Para  $\alpha = 0, 3, \dots, 0, 6$  o arco  $(3, 6)$  é retirado do grafo resultante pois seu grau de possibilidade =  $0, 22$ . A seguir são apresentadas as figuras referentes ao algoritmo:



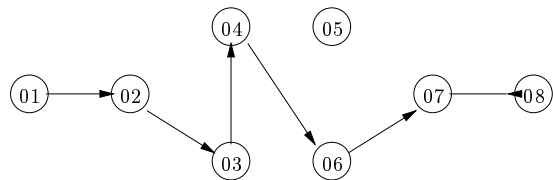
Para  $\alpha = 0, 1$  e  $0, 2$



Para  $\alpha = 0, 3 \dots 0, 6$



Para  $\alpha = 0, 7$  e  $0, 8$



Para  $\alpha = 0, 9$  e  $1, 0$

arco	comprimento	$D_{ij}$	arco	comprimento	$D_{ij}$
(1, 2)	(19, 25, 29)	1, 0	(3, 6)	(54, 57, 62)	0, 2
(2, 3)	(15, 20, 25)	1, 0	(4, 6)	(8, 9, 10)	1, 0
(2, 4)	(58, 63, 68)	0, 8	(5, 7)	(70, 75, 80)	0, 6
(3, 4)	(38, 41, 46)	1, 0	(6, 7)	(65, 75, 85)	1, 0
(4, 5)	(12, 15, 18)	0, 6	(7, 8)	(20, 25, 30)	1, 0

## Referências

- Ahuja, R., Magnati, T. and Orlin, J. (1993). *Network flows : theory, algorithms, and applications*, Prentice-Hall.
- Bazaraa, M. and Jarvis, J. (1990). *Linear programming and network flows*, second edn, John Wiley.
- Dubois, D. and Prade, H. (1980). *Fuzzy Sets and Systems*, Academic Press.
- Klein, C. (1991). Fuzzy shortest path, *Fuzzy Sets and Systems* **39**: 7–41.
- Okada, S. (2001). Interactions among paths in fuzzy shortest path problems, *9th International Fuzzy Systems Association World Congress*, pp. 41–46.
- Okada, S. and Soper, T. (2000). A shortest path problem on a network with fuzzy arc lengths, *Fuzzy Sets and Systems* **109**: 129–140.