

Universidade Estadual de Campinas
Faculdade de Engenharia Elétrica e de Computação
Departamento de Telemática

Contribuições ao Estudo de Grafos Fuzzy: Teoria e Algoritmos

Tese de doutorado apresentada à comissão de pós-graduação da Faculdade de Engenharia Elétrica e de Computação, como requisito para a obtenção do título de DOUTORA em Engenharia Elétrica.

Banca Examinadora:

Autora: Márcia Tomie Takahashi

Orientador: Akebo Yamakami

Akebo Yamakami - FEEC/UNICAMP

Fernando José Von Zuben - FEEC/UNICAMP

Hermano de M. Ferreira Tavares - FEEC/UNICAMP

Laécio Carvalho de Barros - IMECC/UNICAMP

Valéria de Podestá Gomes - IMECC/UNICAMP

Campinas, 10 de maio de 2004

SP - Brasil

Resumo

Neste trabalho, usa-se a teoria de conjuntos fuzzy para modelar incertezas nos parâmetros ou na estrutura de grafos. Na primeira parte do trabalho, realizou-se um minucioso levantamento bibliográfico e determinou-se os problemas a serem estudados, a saber, caminho mínimo fuzzy, árvore geradora mínima fuzzy, emparelhamento fuzzy, casamentos estáveis fuzzy, fluxo máximo fuzzy e problema de fluxo de custo mínimo fuzzy. Cada capítulo da segunda parte deste trabalho apresenta a teoria, algoritmos e exemplos para cada um dos problemas estudados.

Abstract

We apply the fuzzy theory to model uncertainties arising in the parameters of a graph or in its structure. In the first part of this work we performed a survey on the topic and determined the problems to be studied, namely, fuzzy shortest path, fuzzy minimum spanning tree, fuzzy matching, fuzzy marriage, fuzzy maximum flow and minimum cost flow problems. The second part of this work comprises the theory, algorithms and examples for each one of the studied problems.

Dedico aos meus avós,
Kojiro (*in memorian*), Kyio
Takeo e Shizue (*in memorian*)

Por me mostrarem que um oceano só é um obstáculo
se você acredita que os seus sonhos
não valem a pena.

Agradecimentos

Ao meu orientador, Akebo, pela oportunidade dada e pelos ótimos conselhos e sugestões não somente para este projeto como para a minha vida acadêmica.

À minha família, que mesmo com todas as dificuldades que passamos todos estes anos, compreenderam meu esforço e sempre me apoiaram.

Ao Edilson, pela colaboração com o texto, seja qual for o idioma, pela paciência e pelo companheirismo.

Aos meus amigos. O pessoal da faculdade, cada um com sua própria estória, com quem eu dividi esta experiência.

Ao CNPq e à FAPESP, pelo suporte financeiro fornecido durante toda a etapa de estudo e conclusão do doutorado.

Sumário

1	Introdução	1
PARTE I: TEORIA		
2	Conceitos básicos	5
2.1	Conjuntos fuzzy	5
2.2	Tipos de função de pertinência	6
2.2.1	Função triangular	6
2.2.2	Função Γ	7
2.2.3	Função trapezoidal	7
2.2.4	Função Gaussiana	7
2.3	Conceitos básicos de conjuntos fuzzy	7
2.4	Relações fuzzy sobre conjuntos fuzzy	8
2.5	Composição de Relações Fuzzy	9
2.6	Reflexividade e Simetria	10
2.7	Teoria de Possibilidade	11
3	Conceitos em Grafos Fuzzy	13
3.1	Grafos Fuzzy	13
3.2	Caminhos e Conectividade	15
3.3	Clusters	17
3.4	Pontes e Nós de Corte	18
3.5	Florestas e Árvores	23
3.6	Cliques Fuzzy	33
3.7	Ciclos fuzzy e cociclos fuzzy	35
3.8	Métrica em Grafos Fuzzy	41
3.9	Grupos fuzzy de um grafo fuzzy	43

4	Problemas de grafos fuzzy e suas soluções	47
4.1	Problemas com estrutura do grafo crisp e parâmetros fuzzy	47
4.1.1	Abordagem de Bellman e Zadeh, 1970	48
4.1.2	Abordagem de Okada, 2001	49
4.2	Problemas com estrutura de grafos fuzzy	52
4.2.1	O trabalho de Delgado et. al, 1990	53

PARTE II: ALGORITMOS

5	Problema de Caminhos Mínimos Fuzzy	57
5.1	Definição do problema do caminho mínimo clássico	57
5.1.1	Formulação do problema do caminho mínimo	58
5.2	O problema de caminho mínimo fuzzy e os estudos encontrados na literatura	59
5.3	Caso I: grafo crisp e parâmetro fuzzy	62
5.3.1	O estudo de Okada (2001 e 2004)	63
5.3.2	O algoritmo de Okada (2001 e 2004): CM-OK04	64
5.3.3	O estudo de Blue <i>et al.</i> (2002)	66
5.3.4	O estudo de Okada e Soper, 2000	67
5.3.5	O algoritmo de Okada e Soper, 2000: CM-OS00	68
5.3.6	Uma proposta de melhoria no algoritmo de Okada	69
5.3.7	Algoritmo modificado (CM-OK04MOD):	70
5.3.8	Complexidade do algoritmo proposto	71
5.3.9	Proposta de um algoritmo genético (CM-AG) para o problema de caminho mínimo com parâmetro fuzzy	72
5.3.10	Exemplos numéricos	74
5.4	Caso II: o problema do caminho mínimo em um grafo fuzzy	76
5.4.1	Algoritmo proposto por Delgado et. al, 1990	77
5.4.2	Complexidade do algoritmo	78
5.4.3	Exemplo Numérico	79
5.4.4	Algumas heurísticas para o problema do caminho mínimo em um grafo fuzzy	80
5.4.5	Encontrar o caminho com maior valor de pertinência: CM-Max	81
5.5	Exemplos numéricos	82

6	Árvore Geradora Mínima Fuzzy	87
6.1	Introdução	87
6.1.1	Definição do caso clássico	88
6.1.2	Estudos encontrados na literatura	89
6.2	Caso I: problema com grafo crisp e parâmetros fuzzy	89
6.2.1	Parte 1: Encontrar a árvore geradora mínima com custo fuzzy de maior grau de possibilidade	91
6.2.2	Parte 2: Encontrar o conjunto fuzzy de soluções do problema da árvore geradora mínima com custo fuzzy	91
6.2.3	Algoritmo AGMCF:	93
6.2.4	Um algoritmo genético para o problema da árvore geradora mínima com parâmetros fuzzy	95
6.2.5	Algoritmo genético AGMCF-AG:	97
6.2.6	Testes e análises	98
6.3	Caso II: O problema da árvore geradora mínima em um grafo fuzzy	99
6.3.1	A adaptação do algoritmo de Delgado et. al., 1990	100
6.3.2	O trabalho de Chunde, 1996	102
6.3.3	Proposta 01: encontrar a árvore geradora fuzzy com maior grau de pertinência	102
6.3.4	Proposta 2: Heurística para obter um subconjunto fuzzy de soluções	104
6.3.5	Heurística (FMST-H1)	104
6.3.6	Exemplos e análises	104
7	Problemas de emparelhamento fuzzy: designação e casamento estável	109
7.1	Definição do caso clássico	109
7.1.1	Estudos encontrados na literatura sobre os problemas de emparelhamento fuzzy	110
7.2	Caso I: Problema de designação com estrutura do grafo crisp e custo fuzzy	111
7.2.1	Algoritmo proposto para o problema de designação com parâmetros fuzzy: PD-PF	112
7.2.2	Algoritmo genético para o problema de designação com parâmetros fuzzy	113
7.2.3	Exemplos e testes	115
7.3	Estudo II: o problema da designação em um grafo fuzzy	117

7.4	Problema do casamento estável	119
7.4.1	Exemplo numérico	122
8	Fluxos Máximo Fuzzy	125
8.1	Revisão Bibliográfica	126
8.2	Problema do Fluxo Máximo Fuzzy	127
8.2.1	Definição do caso clássico	127
8.2.2	Algumas definições do caso do problema com capacidade fuzzy	127
8.2.3	Grafo crisp e capacidade do arco fuzzy: fluxo máximo com valores inteiros	129
8.2.4	Grafo crisp e capacidade do arco fuzzy: fluxo inteiro com restrições inferior e superior de capacidade	130
8.2.5	Grafo crisp e capacidade do arco fuzzy: fluxo contínuo	132
8.2.6	Proposta de algoritmo para o problema de fluxo máximo com capacidade do arco fuzzy	133
8.2.7	Exemplo	136
8.2.8	Caso II: Estrutura do grafo fuzzy	139
8.2.9	Definição de fluxo fuzzy	139
8.2.10	O problema do fluxo máximo em um grafo fuzzy	140
9	Problema de fluxo de custo mínimo fuzzy	141
9.1	Introdução	141
9.1.1	Considerações sobre a incerteza no problema de fluxo de custo mínimo	142
9.2	Problema de fluxo de custo mínimo com incerteza na capacidade dos arcos	143
9.2.1	Algoritmo proposto para o problema de fluxo de custo mínimo com capacidade fuzzy: PFCM01	144
9.3	Problema de fluxo de custo mínimo com custo dos arcos fuzzy	144
9.3.1	Algoritmo	146
9.4	Confiabilidade vs. Otimalidade do conjunto	146
9.4.1	Algoritmo proposto para o problema de fluxo de custo mínimo com custo fuzzy: PFCM02	147
9.4.2	Exemplo	147
10	Conclusões e trabalhos futuros	149

A	Noções sobre a Teoria de Grafos	159
B	Noções sobre algoritmos genéticos	165
B.1	Um esboço do algoritmo genético utilizado neste trabalho	169
C	Dados dos exemplos utilizados	171
C.1	Ali et al., 1999	171
C.2	Okada e Soper, 2000	173
C.3	Um grafo fuzzy completo com 10 nós	175

Lista de Figuras

2.1	Exemplo de uma função característica clássica	6
2.2	Exemplo de uma função de pertinência	6
3.1	Caminhos entre os nós x e z	16
3.2	Exemplo para nós de corte e pontes fuzzy	20
3.3	Exemplo de um grafo sem nós de corte fuzzy	21
3.4	Exemplo de um grafo não separável	21
3.5	Um grafo em que $\mu(x, y)$ não é o máximo apesar de (x, y) ser uma ponte	22
3.6	Grafo fuzzy com mais de um nó de corte	23
3.7	Exemplos de Floresta Fuzzy	24
3.8	Exemplos de grafos que não são florestas	24
3.9	Exemplo de florestas fuzzy	25
3.10	Subgrafos resultantes da figura anterior	25
3.11	Exemplo de floresta fuzzy	26
3.12	(a) e (b): Grafos completos fuzzy	27
3.13	Figura do exemplo 3.5.1	30
3.14	Exemplo de centro em um grafo fuzzy	43
4.1	Exemplo de grafo com parâmetros fuzzy	48
4.2	Exemplo de um gráfico do modelo de Bellman e Zadeh	49
4.3	Exemplo de uma rede fuzzy	49
4.4	Exemplo de árvores geradoras para a Figura 4.3	50
4.5	Comprimento das árvores dadas na Figura 4.4	50
4.6	Exemplo de um grafo com estrutura fuzzy	52
5.1	Exemplo de um problema de caminho mínimo	57
5.2	Exemplo apresentado para a aplicação dos algoritmos de Ford e o de Floyd em um grafo com parâmetros fuzzy	60

5.3	Exemplo da listas de etiquetas	63
5.4	Comparação dos comprimentos fuzzy	66
5.5	Exemplo de dominância de números fuzzy	67
5.6	Grafo máximo para o algoritmo de Okada modificado	71
5.7	Exemplo de codificação para o problema do caminho mínimo fuzzy	73
5.8	Exemplo de crossover para o problema do caminho mínimo fuzzy	73
5.9	Exemplo de mutação para o problema do caminho mínimo fuzzy	73
5.10	Exemplo de uma rede fuzzy:CMex	74
5.11	Exemplo de um problema de caminho mínimo fuzzy	77
5.12	Exemplo de um grafo fuzzy e primeira solução	79
5.13	Comprimento = 70, $\alpha = 0,5$	79
5.14	Comprimento = 75, $\alpha = 0,8$	79
5.15	Grafo Desconexo \Rightarrow FIM	79
6.1	Exemplo de um problema da árvore geradora mínima	88
6.2	Exemplo de um problema da árvore geradora mínima com parâmetros fuzzy	89
6.3	árvores geradoras	90
6.4	Comparação dos comprimentos fuzzy	90
6.5	Arcos que podem substituir a aresta (3,4)	92
6.6	Árvores criadas pelas substituição da aresta (3,4) na árvore T^*	92
6.7	Exemplo de um cromossomo representando uma árvore	96
6.8	Recombinação de duas árvores	96
6.9	Mutação em uma árvore	96
6.10	Solução do problema	98
7.1	Exemplo de um problema de emparelhamento	110
7.2	Exemplo de construção de soluções para o problema	112
7.3	Codificação de uma solução	114
7.4	Inicialização	114
7.5	Crossover	114
7.6	Mutação	115
7.7	Valor de possibilidade da população no primeiro caso	116
7.8	Melhor fitness e média da população para o primeiro caso	117
7.9	Valor de possibilidade da população para o segundo caso	117
7.10	Melhor fitness e média da população para o segundo caso	117
7.11	Exemplo de um grafo bipartido e o grafo G' resultante do α -corte ($\alpha = 0.6$)	118

8.1	Exemplo de um problema do fluxo máximo	127
8.2	Exemplo de um problema do fluxo máximo com capacidade fuzzy com (c_{ij}, u_{ij}) .	133
8.3	Valores dos cortes essenciais do exemplo	133
8.4	Cortes essenciais, região ótima e valor de satisfação proposto por Chanas (1984)	136
9.1	Capacidade fuzzy de um arco	143
9.2	Exemplo de um problema de fluxo de custo mínimo	145
9.3	Soluções do problema anterior: (a), (b), (c)	145
9.4	Outras soluções do exemplo: (d),(e) e (f)	147
A.1	Exemplo de um arco (i, j)	159
A.2	Exemplo de uma aresta $\{i, j\}$	159
A.3	Exemplo de uma árvore geradora	160
A.4	Exemplo de uma cadeia	160
A.5	Exemplo de um caminho	160
A.6	Exemplo de um ciclo em um grafo	160
A.7	Exemplo de um circuito em um grafo	161
A.8	Exemplo de um conjunto de corte $[X, \bar{X}]$	161
A.9	Exemplo de floresta	162
A.10	Exemplo de um grafo bipartido	162
A.11	Exemplo de grafo completo	162
A.12	Exemplo de grafo fortemente e fracamente conectado	163
A.13	Exemplo de um grafo misto	163
A.14	Exemplo de um laço em um grafo orientado	163
A.15	Exemplo de nós capacitados	164
A.16	Exemplo de um sugrafo gerador	164
B.1	Exemplo de cromossomo com codificação binária	166
B.2	Exemplo de mutação simples	167
B.3	Exemplo de mutação indutida em um cromossomo com codificação real	167
B.4	Exemplo de uma população de oito indivíduos	167
B.5	Exemplo de um crossover de um ponto	168
B.6	Exemplo de um crossover de dois pontos	168
B.7	Exemplo de um crossover de dois pontos	168
B.8	Exemplo de um crossover OX com codificação inteira	168
B.9	Exemplo de um algoritmo genético	170

C.1 Rede óptica italiana	172
C.2 Mapa dos Estados Unidos	173

Lista de Tabelas

5.1	Número de comparações realizadas nos testes com os exemplos CMex e italia1 . . .	75
5.2	Comparações feitas no exemplo da rede americana	75
5.3	Testes realizados para o exemplo de Okada e Soper, 2000	82
5.4	Heurística 1: Resolver para um valor de $\alpha = 0,5$. Teste 3: $\alpha = 0,3$	82
5.5	Heurística 2: Determinar o número máximo de partições: $p = 3$	83
5.6	Heurística 3: Solução com melhor relação (valor de pertinência) \times (comprimento do caminho)	83
5.7	Testes com os grafos G500 e G1000	83
5.8	Heurística 1: Resolver para um valor de confiabilidade escolhido: $\alpha = 0,9$	84
5.9	Heurística 2: Determinar o número máximo de partições: $p = 5$	84
5.10	Heurística 3: Solução com melhor relação (valor de pertinência) \times (comprimento do caminho)	84
6.1	Graus de possibilidade obtidos pelos algoritmos ótimo e genético para a rede Itália	99
6.2	Pertinência das arestas no grafo G	105
6.3	Conjunto de soluções, seus custos e valores de pertinência	106
7.1	Peso dos arcos	116
9.1	Grau de possibilidade de cada solução	148
C.1	Distâncias entre as cidades da rede óptica italiana (em km)	171
C.2	Tempo (t , minutos) de percurso entre dois nós (cidades) n_o e n_d	174
C.3	Tabela com os valores de pertinência das arestas no grafo G	175
C.4	Tabela com os custos (pesos) das arestas no grafo G	175

Capítulo 1

Introdução

Dentro da programação matemática clássica existe uma grande quantidade de problemas que podem ser representados na forma de redes, por exemplo: alocação, caminho mínimo, fluxo máximo, etc. com aplicações nas áreas de energia, telecomunicações, computação, transporte, manufatura, dentre outras. Em (Ahuja et al., 1993) e (Bazaraa et al., 1990) pode-se ter uma idéia da importância deste tipo de abordagem, com algoritmos eficientes para resolução dos problemas citados anteriormente. Estes algoritmos são baseados nas propriedades da teoria de grafos (Harary, 1972).

Normalmente, problemas reais têm associados uma série de parâmetros como custo, capacidades, demandas, que não são naturalmente precisos. Além disso, podem existir casos em que nem a estrutura de grafos em si é precisa. Este fato faz com que uma modelagem sob o ponto de vista da teoria de sistemas nebulosos seja muito atraente.

A introdução dos conceitos de *programação matemática fuzzy* remonta ao início da teoria de conjuntos fuzzy, proposta por Zadeh, em 1965. Na última década, houve uma intensificação nos estudos com a extensão das teorias que envolvem a programação linear, não-linear e inteira mista para um ambiente impreciso e o desenvolvimento de novas áreas como a da programação possibilística. Em Inuiguchi e Ramik (2000) são descritos e classificados os problemas de programação matemática fuzzy.

Com os trabalhos apresentados para o tratamento de problemas de programação matemática fuzzy ao longo dos anos, foi constatado que as contribuições não estão associadas apenas à generalização da teoria já consolidada. Diversos avanços significativos e originais têm sido propostos.

Em contraste aos inúmeros trabalhos existentes utilizando a teoria de grafos clássica, o estudo dos grafos fuzzy ainda está na sua fase inicial, tanto na parte teórica quanto em sua aplicação. Isto se deve, em parte, ao fato de que a teoria de grafos, fluxos em redes e os problemas rela-

cionados ser extensa; existem muitos problemas que podem ser tratados utilizando os conceitos da teoria de grafos e de fluxos em redes. Outro motivo se deve às diferentes abordagens de um problema de grafos fuzzy. Em um problema de grafos, podemos ter incertezas tanto na estrutura (nós e/ou arcos) quanto nos parâmetros (custo, capacidade, demanda). Também, quanto à solução dos problemas, foram encontrados diferentes formatos, dado o objetivo proposto em cada trabalho.

Neste trabalho, pretende-se fazer um estudo sobre a teoria de *grafos fuzzy* proposta por Rosenfeld (1975) e da extensão de aplicações da teoria de grafos clássica (*crisp graphs*) para a teoria de conjuntos fuzzy, propondo novos algoritmos para estes problemas e para problemas de fluxo em redes. A teoria de grafos fuzzy já é mencionada em um trabalho de (Takeda, 1973), em que a conectividade de um grafo fuzzy é estudada, mas no trabalho de Rosenfeld (1975) estão as principais definições de grafos fuzzy.

Um observação sobre a teoria de grafos fuzzy é a existência de outra área de estudos sob o nome de *fuzzy graphs*, com trabalhos voltados a área de *metamodelagem*. Zadeh (Zadeh, 1997), (Zadeh, n.d.) define *grafos fuzzy* como uma disjunção de produtos Cartesianos de conjuntos fuzzy. Esta definição também se encontra presente em (Pedrycz e Gomide, 1998). Trabalhos propostos por (Berthold e Huber, 1995), (Huber et al., 1996), (Tan et al., 1998) podem ser úteis para entender as aplicações destes conceitos nos estudos de aproximações relacionais de funções.

Na primeira parte do trabalho, foi feito um estudo sobre a teoria de *grafos fuzzy* proposta inicialmente por (Rosenfeld, 1975) e a extensão da teoria de grafos clássica (*crisp graphs*) para a teoria de conjuntos fuzzy. Para isso, os principais trabalhos referente a teoria de grafos fuzzy foram reunidos, estudados e comparados.

Na segunda parte, foram propostos algoritmos para problemas de grafos e para problemas de fluxos em redes com estrutura do grafo fuzzy ou com parâmetros fuzzy. Durante o estudo realizado na primeira parte do trabalho, estes problemas foram classificados pelo grau de incerteza e divididos em tópicos e capítulos. Alguns algoritmos já haviam sido propostos, como o de fluxos em redes por (Chanas et al., 1995) e o da árvore geradora mínima por (Chunde, 1996). Para o caso do problema com estrutura de grafo fuzzy, os algoritmos foram baseados no algoritmo geral proposto por (Delgado et al., 1990). Estes algoritmos foram tratados caso a caso, definindo parâmetros e α -cortes que seriam utilizados na resolução do problema, generalizando e/ou oferecendo sugestões aos algoritmos encontrados na literatura. Para o caso do problema com estrutura do grafo crisp e parâmetros fuzzy, alguns já haviam sido propostos, outros foram desenvolvidos neste trabalho. Para ambos os casos, heurísticas e meta-heurísticas foram propos-

tas para tratar algumas dificuldades que surgiram durante os estudos, tais como, dimensão dos problemas e do conjunto fuzzy solução, grau de incerteza dos dados, etc.

No capítulo 2, são apresentados os conceitos sobre a teoria de conjuntos fuzzy necessários para a compreensão do trabalho. No capítulo 3 são apresentados os conceitos sobre teoria de grafos fuzzy propostos na literatura. No Capítulo 4 são apresentados os diferentes tipos de problema de grafos fuzzy que podem ser obtidos e diferentes soluções. No Capítulo 5, é abordado o problema do caminho mínimo fuzzy. No Capítulo 6, é apresentado o problema da árvore geradora mínima fuzzy. No Capítulo 7, são apresentados os problemas de emparelhamento fuzzy, no caso o problema de designação e o problema do casamento estável. No Capítulo 8, é apresentado o problema do fluxo máximo. No Capítulo 9, é proposta uma abordagem para o problema de fluxo de custo mínimo. Finalmente, no Capítulo 10, apresentamos as conclusões, outros trabalhos que tratam de problemas com estrutura de grafos em que já foram estudadas incertezas e sugestões de trabalhos futuros.

PARTE I: TEORIA

Capítulo 2

Conceitos básicos

Neste capítulo os conceitos básicos da teoria fuzzy e da teoria de possibilidade serão introduzidos para melhor compreensão desta tese. As referências (Dubois e Prade, 1980) e (Pedrycz e Gomide, 1998) podem ajudar a aprofundar no estudo da teoria de conjuntos fuzzy.

2.1 Conjuntos fuzzy

Conjuntos são uma tentativa de organizar, resumir e generalizar conhecimento sobre objetos. Neste sentido trabalhamos com uma dicotomia sobre um objeto pertencer ou não a um determinado conjunto. Esta dicotomia pode ser representada por uma função característica:

$$f_A(x) = \begin{cases} 1, & \text{se } x \in A \\ 0, & \text{se } x \notin A \end{cases} \quad (2.1)$$

Por exemplo, considere o conjunto dos homens altos $H = \{x \in \mathfrak{R} | x \geq 1,80\}$. Alguém com 1,79m não pode ser considerado alto, ao passo que alguém com 1,81m é seguramente alto.

Um subconjunto fuzzy A em X é definido por uma *função de pertinência* μ_A que associa cada ponto de X a um número real no intervalo $[0,1]$, com o valor de μ_A em x representando o *grau de pertinência* de x em A . Então, quanto mais próximo o valor de $f_A(x)$ estiver da unidade, maior o grau de pertinência de x em A .

Quando houver a necessidade de diferenciar entre os conjuntos clássicos e os conjuntos fuzzy, o conjunto com função característica com dois valores, 0 ou 1, será chamado de conjunto clássico (ordinário ou crisp) ou simplesmente conjunto.

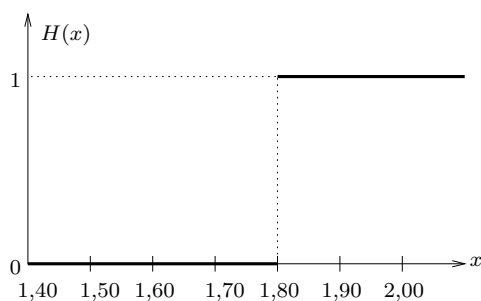


Figura 2.1: Exemplo de uma função característica clássica

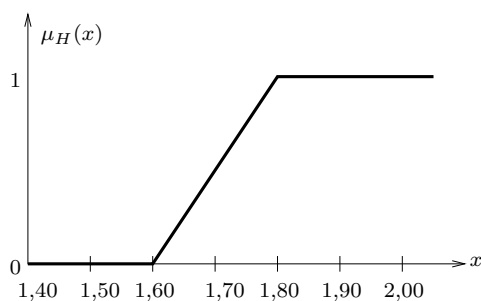


Figura 2.2: Exemplo de uma função de pertinência

2.2 Tipos de função de pertinência

Em princípio, qualquer função do tipo $\mu_S : X \rightarrow [0, 1]$ descreve uma função de pertinência associada ao conjunto fuzzy S . Cada função tem diferentes propriedades e sua utilização pode ser determinada de acordo com o contexto do problema. Aqui, os principais tipos de função de pertinência são apresentados.

2.2.1 Função triangular

$$\mu_S(x) = \begin{cases} 0 & \text{se } x < a \\ \frac{x-a}{m-a} & \text{se } x \in [a, m] \\ \frac{b-x}{b-m} & \text{se } x \in [m, b] \\ 0, & \text{se } x > b \end{cases} \quad (2.2)$$

sendo m o valor modal e a e b os limitantes inferior e superior, respectivamente.

2.2.2 Função Γ

$$\mu_S(x) = \begin{cases} 0 & \text{se } x \leq a \\ 1 - \exp\{-k(x - a)^2\}, & \text{se } x > a \end{cases} \quad (2.3)$$

com $k > 0$.

2.2.3 Função trapezoidal

$$\mu_S(x) = \begin{cases} 0 & \text{se } x < a \\ \frac{x-a}{m-a} & \text{se } x \in [a, m] \\ 1 & \text{se } x \in [m, n] \\ \frac{b-x}{b-n} & \text{se } x \in [n, b] \\ 0, & \text{se } x > b \end{cases} \quad (2.4)$$

2.2.4 Função Gaussiana

$$S(x) = \exp\{-k(x - m)^2\} \quad (2.5)$$

para $k > 0$.

2.3 Conceitos básicos de conjuntos fuzzy

Aqui estão alguns conceitos básicos dos conjuntos fuzzy que serão necessários no decorrer do texto:

Definição 2.3.1. *Um conjunto fuzzy S é normal se sua função de pertinência possui pelo menos um valor tal que $\mu_S(x_o) = 1$.*

Caso não exista um valor x tal que o supremo da função de pertinência seja igual a um, então S é subnormal. O supremo é considerado a *altura* de S .

Definição 2.3.2. *O suporte de um subconjunto fuzzy S , denotado por $\text{supp}(S)$ é dado por:*

$$\text{supp}(S) = \{x \in X | \mu_S(x) > 0\}$$

ou seja, o suporte é formado pelos elementos que possuem valores de pertinência não-nulos.

Definição 2.3.3. O núcleo de um conjunto fuzzy S é o conjunto de todos os elementos cujo valor de pertinência é igual a 1

$$\text{nucleo}(S) = \{x \in X | \mu_S(x) = 1\}$$

Definição 2.3.4. Um conjunto fuzzy S é convexo se sua função de pertinência é tal que

$$\mu_S[\lambda x_1 + (1 - \lambda)x_2] \geq \min[\mu_S(x_1), \mu_S(x_2)]$$

para quaisquer x_1 e $x_2 \in X$ e $\lambda \in [0, 1]$.

2.4 Relações fuzzy sobre conjuntos fuzzy

Seja X um conjunto. Um subconjunto fuzzy de X possui uma função $\sigma : S \rightarrow [0, 1]$ que associa elementos $x \in X$ a um grau de pertinência, $0 \leq \sigma(x) \leq 1$. Similarmente, uma *relação fuzzy* sobre X é um subconjunto fuzzy de $X \times X$, que possui uma função $\mu : S \times X \rightarrow [0, 1]$ que associa cada par ordenado de elementos (x, y) a um grau de pertinência $0 \leq \mu(x, y) \leq 1$. Nos casos especiais onde σ e μ podem somente assumir valores 0 e 1, eles se tornam funções características de um subconjunto ordinário de X e uma relação ordinária sobre X , respectivamente.

Se $T \subseteq X$ é um subconjunto de X e $R \subseteq X \times X$ é uma relação sobre X , então R é uma relação sobre T contanto que $(x, y) \in R$ implique que $x \in T$ e $y \in T$ para todo x, y . Sejam τ e ρ funções características de T e R , respectivamente. Então esta condição pode ser estabelecida como

$$\rho(x, y) = 1 \text{ implica que } \tau(x) = \tau(y) = 1, \quad \forall x, y \in S$$

Podemos associar estas condições às funções de pertinência:

$$\rho(x, y) \leq \tau(x) \bigwedge \tau(y), \quad \forall x, y \in S$$

onde \bigwedge significa *ínfimo*.

Retomando o caso geral onde σ é um subconjunto fuzzy de S e μ uma relação fuzzy sobre S , diz-se que μ é uma *relação fuzzy sobre σ* se

$$\mu(x, y) \leq \sigma(x) \bigwedge \sigma(y), \quad \forall x, y \in S.$$

Em outras palavras, para μ ser uma relação fuzzy sobre σ , é necessário que o grau de pertinência de um par de elementos nunca exceda o grau de pertinência dos próprios elementos.

Se pensarmos nos elementos como nós em um grafo e os pares como os arcos (veja seção 3.1), isto equivale ao requisito de que a força de um arco nunca exceda as forças de seus nós.

Proposição 2.4.1. (Rosenfeld, 1975) Para um dado subconjunto fuzzy σ de S , a relação mais forte sobre S , ou seja, uma relação fuzzy sobre σ , é μ_σ , definida por

$$\mu_\sigma(x, y) = \sigma(x) \wedge \sigma(y), \quad \forall x, y \in S$$

Proposição 2.4.2. (Rosenfeld, 1975) Para uma dada relação fuzzy μ sobre S , o subconjunto fuzzy mais fraco de S para o qual μ é uma relação fuzzy é σ_μ , definido por

$$\sigma_\mu(x) = \sup_{y \in S} [\mu(x, y) \vee \mu(y, x)], \quad \forall x \in S$$

onde \vee significa supremo.

Para algum limiar t , $0 \leq t \leq 1$, o conjunto

$$\sigma_t = \{x \in S \mid \sigma(x) \geq t\}$$

é um subconjunto de S e o conjunto

$$\mu_t = \{(x, y) \in S \times S \mid \mu(x, y) \geq t\}$$

é uma relação sobre S . Usando esta notação, podemos estabelecer:

Proposição 2.4.3. (Rosenfeld, 1975) Seja μ uma relação fuzzy sobre σ e seja $0 \leq t \leq 1$. Então μ_t é uma relação sobre σ_t .

2.5 Composição de Relações Fuzzy

Seja μ e ν relações sobre σ . Pela composição de μ e ν é entendido o conjunto $\mu \circ \nu$ definido por

$$(\mu \circ \nu)(x, z) = \sup_{y \in S} [\mu(x, y) \wedge \nu(y, z)], \quad \forall x, z \in S$$

Outras definições de composição de relações fuzzy serão discutidas a seguir. Esta definição é chamada composição max-min.

Proposição 2.5.1. (Rosenfeld, 1975) $\mu \circ \nu$ é uma relação fuzzy sobre σ .

É bem conhecido que a composição de relações fuzzy é associativa, isto é, $\forall \mu, \nu, \rho$ temos $\mu \circ (\nu \circ \rho) = (\mu \circ \nu) \circ \rho$. Podemos então definir a potência de uma relação fuzzy como $\mu^1 = \mu$; $\mu^2 = \mu \circ \mu$; $\mu^3 = \mu \circ \mu^2 = \mu \circ \mu \circ \mu$ e assim por diante. Também podemos definir

$$\mu^\infty = \sup_{k=1,2,\dots} \mu^k.$$

Finalmente, é conveniente definir

$$\begin{aligned} \mu^0(x, y) &= 0 \quad \text{se } x \neq y \\ \mu^0(x, x) &= \sigma(x) \end{aligned}$$

para todo $x, y \in S$.

Proposição 2.5.2. (Rosenfeld, 1975) Para todo t , $0 \leq t \leq 1$, temos $(\mu \circ \nu)_t = \mu_t \circ \nu_t$.

Proposição 2.5.3. (Rosenfeld, 1975) Se $\mu \leq \nu$ e $\lambda \leq \rho$, então $\mu \circ \lambda \leq \nu \circ \rho$.

2.6 Reflexividade e Simetria

Seja μ uma relação fuzzy sobre σ . μ é reflexivo se

$$\mu(x, x) = \sigma(x), \quad \forall x \in S$$

Proposição 2.6.1. (Rosenfeld, 1975) Se μ é reflexivo, então temos que $\mu(x, y) \leq \mu(x, x)$ e $\mu(y, x) \leq \mu(x, x)$, $\forall x, y \in S$

Proposição 2.6.2. (Rosenfeld, 1975) Se μ é uma relação fuzzy reflexiva sobre σ , então para $0 \leq t \leq 1$ qualquer, μ_t é uma relação reflexiva sobre σ_t .

Corolário 2.6.1. (Rosenfeld, 1975) Se μ é reflexivo, $\mu \leq \mu \circ \mu$.

Corolário 2.6.2. (Rosenfeld, 1975) Se μ é reflexivo, $\mu^0 \leq \mu^1 \leq \mu^2 \leq \dots \leq \mu^\infty$.

Corolário 2.6.3. (Rosenfeld, 1975) Se μ é reflexivo, $\mu^0(x, x) = \mu^1(x, x) = \mu^2(x, x) = \dots = \mu^\infty(x, x) = \sigma(x)$

Diz-se que μ é simétrico se $\mu(x, y) = \mu(y, x)$ para todo $x, y \in S$. É claro que se μ é simétrico, μ_t também é para qualquer limiar t . Note que a propriedade de simetria não depende da escolha do subconjunto fuzzy σ , diferente da reflexividade.

Proposição 2.6.3. (Rosenfeld, 1975) Se μ e ν são simétricos, então $\mu \circ \nu$ é simétrico se, e somente se, $\mu \circ \nu = \nu \circ \mu$.

Corolário 2.6.4. (Rosenfeld, 1975) Se μ é simétrico, então todas as potências de μ também são.

Estes últimos resultados pertencem a qualquer definição de composição que é baseada em operações comutativas em $[0,1]$.

2.7 Teoria de Possibilidade

Seja um grafo $G = (N, A)$ com custo associado $\tilde{c}, \in \mathfrak{R}^n$. Sejam dois números fuzzy \tilde{T}^1 e \tilde{T}^2 , $\tilde{T}^1 \neq \tilde{T}^2$. Podemos dizer que T^1 tem um grau de possibilidade de ser menor do que T^2 dado por (Okada, 2001):

$$\tilde{w} = Poss\left(\sum_{ij \in T^1} \tilde{c}_{ij} \leq \sum_{ij \in T^2} \tilde{c}_{ij}\right) = \sup_{u \leq v} \min\{\mu_{T^1}(u), \mu_{T^2}(v)\} \quad (2.6)$$

Esta equação também é estudada em (Dubois e Prade, 1980).

Para encontrar uma solução fuzzy utilizando a teoria de possibilidade, teríamos que encontrar todas as soluções que possuem algum grau de possibilidade de ser a solução ótima e comparar estas soluções para obter o grau de possibilidade de cada uma (Okada, 2001). O grau de possibilidade é dado pela fórmula:

$$D_T = \min_{T^k \in \mathcal{T}} \left\{ Poss\left(\sum_{ij \in T^k} c_{ij} \leq \sum_{ij \in \mathcal{T}} c_{ij}\right) \right\} \quad (2.7)$$

e o grau de possibilidade de cada arco é dado pela equação

$$D_{ij} = \max_{T^k | ij \in T^k} \{D_{T^k}\} \quad (2.8)$$

Isso torna o problema de difícil resolução pois, além de ter que enumerar todas as soluções, a comparação entre elas torna o problema NP-completo.

Proposição 2.7.1. Sejam dois números fuzzy, normalizados, \tilde{a} e \tilde{b} . Temos que, se

$$\inf_{u | \mu_{\tilde{a}}(u)=1} \{u\} \leq \sup_{v | \mu_{\tilde{b}}(v)=1} \{v\}, \text{ então } Poss(a \leq b) = 1$$

Prova: Seja a definição do grau de possibilidade entre dois números fuzzy:

$$Poss(a \leq b) = \sup_{u \leq v} \{\min\{\mu_a(u), \mu_b(v)\}\}$$

Suponha que $u = \inf_{u|\mu(u)=1}\{u\}$ e $v = \sup_{v|\mu(v)=1}\{v\}$, então $u \leq v$ é satisfeito. Com isso, $\mu_a(u) = \mu_b(v) = 1 \Rightarrow \min\{\mu_a(u), \mu_b(v)\} = 1$. Portanto, $Poss(a \leq b) = 1$. \square

Teorema 2.7.1. *Seja T^0 uma solução do grafo crisp G_C , com valores modais de G sendo os custos do grafo crisp. Seja a solução fuzzy \tilde{T}^* , associada à solução crisp T^0 , com custo igual a $\tilde{w}^* = \sum_{ij \in \tilde{T}^*} \tilde{c}_{ij}$. Se \tilde{T}^* for a solução ótima do problema, então $Poss(\sum_{ij \in \tilde{T}^*} \tilde{c}_{ij} \leq \sum_{ij \in \tilde{T}^k} \tilde{c}_{ij}) = 1$, sendo \tilde{T}^k qualquer solução em G .*

Prova: Suponha que exista um T^k tal que $Poss(\sum_{ij \in T^*} c_{ij} \leq \sum_{ij \in T^k} c_{ij}) < 1$. Pela proposição 2.7.1, temos que $w_{c^*} > w_{c_k}$. Mas, como w_c é o menor valor encontrado para o grafo G_C , então $w_{c^*} \leq w_k$ chegando a uma contradição. Portanto, $Poss(\sum_{ij \in T^*} c_{ij} \leq \sum_{ij \in T^k} c_{ij}) = 1$. \square

Proposição 2.7.2. *Sejam \tilde{a} e \tilde{b} dois números fuzzy. Se $\sup\{supp\{\tilde{b}\}\} < \inf\{supp\{\tilde{a}\}\}$, então $Poss(a \leq b) = 0$.*

Prova: Seja um número fuzzy \tilde{a} . Temos que $\mu(u) = 0$, se $u \notin (a, \bar{a})$, onde a e \bar{a} são os limitantes inferior e superior, respectivamente, do número fuzzy. Se $u \in [a, \bar{a}]$, então $\mu(u) \geq 0$. Como $\bar{b} = \sup\{supp\{\tilde{b}\}\} < \inf\{\{\tilde{a}\}\} = a$, então

$$u \geq a > \bar{b} \geq v \Rightarrow \nexists v \in (b, \bar{b}) | u \leq v \Rightarrow \mu_{\bar{b}}(v) = 0$$

Portanto

$$\min\{\mu_{\bar{a}}(u), \mu_{\bar{b}}\} = 0, \forall u, v \Rightarrow Poss(a \leq b) = 0$$

\square

Capítulo 3

Conceitos em Grafos Fuzzy

Com base nos conceitos apresentados no Capítulo 2, vamos estabelecer os conceitos básicos referentes à teoria de grafos fuzzy. Para ver os conceitos básicos da teoria de grafos clássica, veja o Apêndice ?? ou, para um estudo mais aprofundado da teoria de grafos clássica, veja (Ahuja et al., 1993).

3.1 Grafos Fuzzy

Qualquer relação $R \subseteq S \times S$ sobre um conjunto S pode ser vista como definindo um *grafo* com conjunto de nós S e conjunto de arcos R . Similarmente, qualquer relação fuzzy $\mu : S \times S \rightarrow [0, 1]$ pode ser considerada um *grafo ponderado* ou *grafo fuzzy*, onde o nó $x \in S$ tem peso $\sigma(x) \in [0, 1]$ e o arco $(x, y) \in S \times S$ tem peso $\mu(x, y) \in [0, 1]$. Nesta e nas seções seguintes devemos utilizar terminologia da teoria de grafos e introduzir analogias de diversos conceitos básicos desta teoria. Por questão de simplicidade, consideraremos somente grafos não-direcionados, isto é, a relação fuzzy é simétrica. Com isso, todos os arcos podem ser vistos como pares não ordenados de nós. Não serão considerados laços, isto é, arcos da forma (x, x) . Poderemos assumir, se desejarmos, que a relação fuzzy será reflexiva.

Definimos $G^* = (\sigma^*, \mu^*)$ como o grafo crisp associado ao grafo fuzzy $G = (\sigma, \mu)$ onde $\sigma^* = \{x \in S | \sigma(x) > 0\}$ e $\mu^* = \{x, y \in S \times S | \mu(x, y) > 0\}$.

Formalmente, um grafo fuzzy $G = (\sigma, \mu)$ é um par de funções $\sigma : S \rightarrow [0, 1]$ e $\mu : S \times S \rightarrow [0, 1]$, onde para todo $x, y \in S$ temos $\mu(x, y) \leq \sigma(x) \wedge \sigma(y)$. O grafo fuzzy $H = (\tau, \nu)$ é chamado um

subgrafo fuzzy (parcial) de G se

$$\tau(x) \leq \sigma(x), \quad \forall x \in S$$

e

$$\nu(x, y) \leq \mu(x, y), \quad \forall x, y \in S$$

Para qualquer limiar t , com $0 \leq t \leq 1$, se $\sigma_t = \{x \in S | \sigma(x) \geq t\}$ e $\mu_t = \{(x, y) \in S \times S | \mu(x, y) \geq t\}$, então temos $\mu_t \subseteq \sigma_t \times \sigma_t$, fazendo com que (σ_t, μ_t) seja um grafo com o conjunto de nós σ_t e o conjunto de arcos μ_t .

Proposição 3.1.1. (Rosenfeld, 1975) Se $0 \leq u \leq v \leq 1$, então (σ_v, μ_v) é um subgrafo de (σ_u, μ_u) .

Prova: Seja (σ_v, μ_v)

$$\sigma_v = \{x \in S | \sigma(x) \geq v\} \quad \text{e} \quad \mu_v(x, y) = \{(x, y) \in S \times S | \mu(x, y) \geq v\}$$

e (σ_u, μ_u)

$$\sigma_u = \{x \in S | \sigma(x) \geq u\} \quad \text{e} \quad \mu_u(x, y) = \{(x, y) \in S \times S | \mu(x, y) \geq u\}$$

como $v \geq u$, todos os elementos de (σ_v, μ_v) também pertencem a (σ_u, μ_u) , ou seja, para todo $x \in \sigma_v$, $\sigma(x) \geq v \geq u \Rightarrow x \in \sigma_u$ ($\sigma_v(x) = \sigma_u(x)$, $x \in \sigma_v$) e para todo $(x, y) \in \mu_v$, $\mu(x, y) \geq v \geq u \Rightarrow (x, y) \in \mu_u$ ($\mu_v(x, y) = \mu_u(x, y)$, $(x, y) \in \mu_v$).

Portanto

$$(\sigma_v, \mu_v) \subseteq (\sigma_u, \mu_u).$$

□

Proposição 3.1.2. (Rosenfeld, 1975) Se (τ, ν) é um subgrafo fuzzy de (σ, μ) , então para qualquer limiar t , $0 \leq t \leq 1$, (τ_t, ν_t) é um subgrafo de (σ_t, μ_t) .

Prova: Sendo (τ, ν) um subgrafo fuzzy de (σ, μ) temos

$$\tau(x) \leq \sigma(x), \quad \forall x \in S$$

e

$$\nu(x, y) \leq \mu(x, y), \quad \forall (x, y) \in S \times S$$

Seja (τ_t, ν_t) e (σ_t, μ_t) . Temos

$$\sigma(x) \geq \tau(x) \geq t, \quad \forall x \in \tau_t$$

e

$$\mu(x, y) \geq \nu(x, y) \geq t, \forall (x, y) \in \nu_t$$

Portanto, (τ_t, ν_t) é um subgrafo fuzzy de (σ_t, μ_t) para $0 \leq t \leq 1$. □

Definição 3.1.1. (Rosenfeld, 1975) *Seja (τ, ν) um subgrafo fuzzy de (σ, μ) . Então (τ, ν) é um subgrafo gerador fuzzy de (σ, μ) se, e somente se, $\tau(x) = \sigma(x), \forall x$.*

Neste caso, os dois grafos têm o mesmo conjunto de nós e diferem somente no grau de pertinência dos arcos.

Para algum subconjunto fuzzy τ de σ , isto é, tal que $\tau(x) \leq \sigma(x)$ para todo x , o subgrafo fuzzy de (σ, μ) induzido por τ é o subgrafo fuzzy máximo de (σ, μ) que tem o conjunto de nós τ . Evidentemente, este é justamente o grafo (τ, ν) , onde

$$\nu(x, y) = \tau(x) \wedge \tau(y) \wedge \sigma(x, y), \quad \forall x, y \in S$$

Consideraremos agora que o conjunto base S de um grafo fuzzy é sempre finito.

3.2 Caminhos e Conectividade

Um *caminho* ρ em um grafo fuzzy é uma seqüência de nós distintos x_0, x_1, \dots, x_n tal que $\mu(x_{i-1}, x_i) > 0, 1 \leq i \leq n$; aqui $n \geq 0$ é chamado de *comprimento* de ρ (número de arestas que pertencem ao caminho). Pares consecutivos (x_{i-1}, x_i) são chamados de arcos do caminho. A *força* de ρ é definida como $\bigwedge_{i=1}^n \mu(x_{i-1}, x_i)$. Em outras palavras, a força de um caminho é definida como sendo o peso do seu arco mais fraco. Se o caminho tem comprimento 0, é conveniente definir sua força como sendo $\sigma(x_0)$. Chamamos ρ um ciclo se $x_0 = x_n$ e $n \geq 3$.

Definição 3.2.1. (Rosenfeld, 1975)

1. (σ, μ) é um ciclo se, e somente se, (σ^*, μ^*) é um ciclo.
2. (σ, μ) é um ciclo fuzzy se, e somente se, (σ^*, μ^*) é um ciclo e $\nexists x \in \mu^*$ único tal que $\mu(x) = \min\{\mu(y) | y \in \mu^*\}$.

Dois nós que estão unidos por um caminho (isto é, o primeiro e o último nó) são ditos conectados. É evidente que *conectividade* é uma relação reflexiva e simétrica, e que também é transitiva. De fato, x e y estão conectados se, e somente se, $\mu^\infty(x, y) > 0$. As classes equivalentes de nós sob esta relação são chamadas *componentes conectados* do grafo fuzzy dado; elas estão em seus subgrafos fuzzy conectados máximos. Um *caminho mais forte* unindo quaisquer dois nós x, y tem força $\mu^\infty(x, y)$; podemos às vezes referir a este como sendo a força de conectividade entre os nós.

Proposição 3.2.1. (Rosenfeld, 1975) *Se (τ, ν) é um subgrafo fuzzy de (σ, μ) , então para todo $x, y \in S$ temos $\nu^\infty(x, y) \leq \mu^\infty(x, y)$.*

Prova: Como conectividade é uma relação reflexiva e transitiva, temos pelo Corolário 2.6.3 que $\mu = \mu^k$, $k = 1, 2, \dots$. Então

$$\nu^\infty(x, z) = \sup_{k=1,2,\dots} \nu^k(x, z) \leq \sup_{k=1,2,\dots} \mu^k(x, z) = \mu^\infty(x, z), \forall x, z \in S$$

□

Pode-se definir a *distância* entre os nós x e y como o comprimento do caminho mais forte e mais curto entre estes nós. Esta *distância* é, de fato, simétrica e definida positiva (pela definição de um grafo fuzzy, nenhum caminho de x para x pode ter mais força que $\sigma(x)$, que é a força do caminho de comprimento 0). Contudo, esta distância não é triangular, como podemos observar no exemplo da Figura 3.1 a seguir:

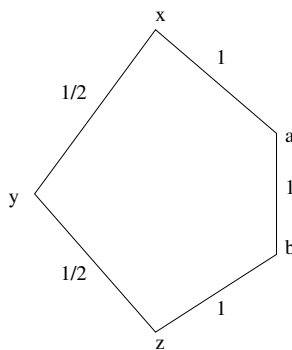


Figura 3.1: Caminhos entre os nós x e z

Aqui, qualquer caminho de x para y ou de y para z tem força $\leq \frac{1}{2}$, já que eles devem conter ou o arco (x, y) ou o (y, z) . Assim, os caminhos mais curtos e mais fortes entre estes nós têm comprimento 1. Por outro lado, existe um caminho de x para z , passando por a e b , que tem comprimento 3 e força 1. Assim $d(x, z) = 3 > 1 + 1 = d(x, y) + d(y, z)$, neste caso.

Uma melhor noção de distância em um grafo fuzzy pode ser definida como: para qualquer $\rho = (x_0, \dots, x_n)$ define-se o μ -comprimento de ρ como a soma dos inversos dos pesos dos arcos de ρ , isto é,

$$l(\rho) = \sum_{i=1}^n \frac{1}{\mu(x_{i-1}, x_i)}$$

Se $n = 0$, definimos $l(p) = 0$. Obviamente, para $n \geq 1$ temos $l(p) \geq 1$. Para quaisquer dois nós x e y , podemos definir sua μ -distância $\delta(x, y)$ como o menor μ -comprimento de todos os caminhos de x para y .

3.3 Clusters

Em grafos crisps existem diversos modos de definir *clusters* de nós. Uma abordagem é chamar o conjunto C de nós de *cluster* de ordem k se

- (a) Para todos os nós $x, y \in C$ temos $d(x, y) \leq k$
- (b) Para todos os nós $z \notin C$ temos $d(z, w) > k$ para qualquer $w \in C$.

onde $d(a, b)$ é o comprimento do caminho mais curto entre a e b . Em outras palavras, em um *cluster* C de ordem K , todo par de nós está a uma distância máxima de k de cada nó e C é máximo com respeito a esta propriedade, isto é, nenhum nó fora de C está dentro da distância k para todo nó em C .

Quando $k = 1$, um k -cluster é chamado um *clique*, um subgrafo completo máximo, isto é, um subgrafo máximo em que cada par de nós é unido por um arco. No outro extremo, seja $k \rightarrow \infty$, um k -cluster torna-se um *componente conectado*, ou seja, um subgrafo máximo no qual cada par de nós está unido por um caminho (de qualquer comprimento).

Estas idéias podem ser generalizadas para grafos fuzzy como a seguir: em $G = (\sigma, \mu)$, podemos chamar $C \subseteq S$ um *cluster fuzzy* de ordem k se

$$\inf_{x, y \in C} \mu^k(x, y) > \sup_{z \notin C} (\inf_{w \in C} \mu^k(w, z))$$

Note que C é um subconjunto crisp de S , não um subconjunto fuzzy. Se G é um grafo crisp, temos $\mu^k(a, b) = 0$ ou 1 para todo a e b ; por isso esta definição se resume a

- (a) $\mu^k(x, y) = 1$ para todo $x, y \in C$
- (b) $\mu^k(w, z) = 0$ para todo $z \notin C$ e para algum $w \in C$

De fato, os k -clusters obtidos usando esta definição são justamente cliques clássicos em grafos obtidos tolerando os limites da k -ésima potência do grafo fuzzy dado. Seja C um k -cluster fuzzy e seja $\inf_{x,y \in C} \mu^k(x,y) = t$. Se levarmos μ^k (e σ) a t , obteremos um grafo crisp no qual C é agora um clique crisp.

3.4 Pontes e Nós de Corte

Seja $G = (\sigma, \mu)$ um grafo fuzzy e sejam x, y dois nós distintos quaisquer, e seja G' um subgrafo fuzzy de G obtido pela retirada do arco (x, y) , isto é $G' = (\sigma, \mu')$ onde

$$\begin{aligned} \mu'(x, y) &= 0; \\ \mu'(x, y) &= \mu \quad \text{para todos os outros pares.} \end{aligned} \tag{3.1}$$

Definição 3.4.1. (Rosenfeld, 1975). Um arco (x, y) é uma ponte fuzzy de $G = (\sigma, \mu)$ se $\mu'^\infty(u, v) < \mu^\infty(u, v)$ para algum u, v . Em outras palavras, (x, y) é uma ponte se a retirada de (x, y) reduz a força de conectividade entre alguns pares de nós.

Equivalentemente, (x, y) é uma ponte fuzzy se, e somente se, existem nós u e v tais que (x, y) é um arco pertencente a todos os caminhos mais fortes entre u e v .

Teorema 3.4.1. (Rosenfeld, 1975) As seguintes declarações são equivalentes

- (a) (x, y) é uma ponte
- (b) $\mu'(x, y) < \mu(x, y)$
- (c) (x, y) não é o arco mais fraco de nenhum ciclo

Prova: Se (x, y) não é uma ponte, devemos ter $\mu'^\infty(x, y) = \mu^\infty(x, y) \geq \mu(x, y)$. Assim (b) implica (a). Se (x, y) é um arco mais fraco de um ciclo, então todos os caminhos envolvendo o arco (x, y) podem ser convertidos em caminho que não contém este arco, pelo menos tão forte, usando o resto do ciclo como um caminho de x para y ; assim (x, y) não pode ser uma ponte, portanto (a) implica (c). Se $\mu'^\infty(x, y) \geq \mu(x, y)$, existe um caminho de x para y não envolvendo (x, y) , que tem força $\geq \mu(x, y)$ e este caminho junto com (x, y) forma um ciclo no qual (x, y) é o arco mais fraco. Assim (c) implica em (b). \square

Observação 3.4.1. (Sunitha e Vijayakumar, 1999) Seja $G = (\sigma, \mu)$ um grafo fuzzy tal que o grafo crisp associado seja um ciclo e seja $t = \min\{\mu(u, v) \mid \mu(u, v) > 0\}$. Então todos os arcos (u, v) tal que $\mu(u, v) > t$ são pontes fuzzy de G .

Seja w qualquer nó, e seja G^* o subgrafo fuzzy de G obtido pela retirada do nó w , isto é, G^* é o subgrafo induzido por σ^* , onde

$$\begin{aligned}\sigma^*(w) &= 0 \\ \sigma^* &= \sigma \quad \text{para todos os outros nós.}\end{aligned}$$

Definição 3.4.2. (Rosenfeld, 1975). Um nó é um nó de corte fuzzy de $G = (\sigma, \mu)$ se

$$\mu^{*\infty}(u, v) < \mu^\infty(u, v)$$

para algum u, v (diferente de w). Em outras palavras, se a remoção deste nó reduz a força de conectividade entre alguns pares de outros nós.

Equivalentemente, w é um nó de corte fuzzy se, e somente se, existem u, v distintos de w tais que w pertença a todos os caminhos mais fortes entre u e v .

Teorema 3.4.2. (Sunitha e Vijayakumar, 1999) Seja $G = (\sigma, \mu)$ um grafo fuzzy tal que o grafo crisp associado $G^* = (\sigma^*, \mu^*)$ seja um ciclo. Então um nó é um nó de corte fuzzy de G se, e somente se, este é um nó comum a duas pontes fuzzy.

Prova: Seja w um nó de corte fuzzy em G . Então existem u e v , distintos de w , tais que w está em todos os caminhos mais fortes entre u e v . Mas sendo $G^* : (\sigma^*, \mu^*)$ um ciclo, existe somente um caminho mais forte entre u e v contendo w e, pela Observação 3.4.1, todos estes arcos são pontes fuzzy. Então w é um nó comum de duas pontes fuzzy. Inversamente, seja w um nó comum de duas pontes fuzzy (u, w) e (w, v) . Então ambos (u, w) e (w, v) não são os arcos mais fracos de G (Teorema 3.4.1). Também o caminho de u e v que não contém os arcos (u, w) e (w, v) tem força menor que $\mu(u, w) \wedge \mu(w, v)$. Portanto w é um nó de corte fuzzy. \square

Teorema 3.4.3. (Sunitha e Vijayakumar, 1999) Se w é um nó comum de pelo menos duas pontes fuzzy, então w é um nó de corte fuzzy.

Prova: Sejam (u_1, w) e (w, u_2) duas pontes fuzzy. Então existem u e v tais que (u_1, w) pertence a todos os caminhos mais fortes entre u e v . Se w é distinto de u e v , segue que w é um nó de corte fuzzy. A seguir, suponha que u ou v seja o nó w ; com isso (u_1, w) está em todo caminho mais forte de $u - w$ ou (w, u_2) está em todo caminho mais forte de w e v . Se possível, suponha que w não seja um nó de corte fuzzy. Então entre todos os pares de nós existe pelo menos um caminho mais forte que não contém w . Em particular, existe pelo menos um caminho mais forte ρ , unindo u_1 e u_2 que não contém w . Este caminho junto a (u_1, w) e (w, u_2) forma um ciclo.

Caso 1: Se u_1, w, u_2 não é um caminho mais forte, então claramente ou (u_1, w) ou (w, u_2) ou ambos tornam-se os arcos mais fracos do ciclo, o que contradiz que (u_1, w) e (w, u_2) são pontes fuzzy.

Caso 2: Se u_1, w, u_2 também é um caminho mais forte unindo u_1 a u_2 , então seja $\mu^\infty(u_1, u_2) = \mu(u_1, w) \wedge \mu(w, u_2)$ a força de ρ . Portanto arcos de ρ são pelo menos tão fortes como $\mu(u_1, w)$ e $\mu(w, u_2)$, o que implica que (u_1, w) , (w, u_2) ou ambos são os arcos mais fracos do ciclo, o que novamente é uma contradição. \square

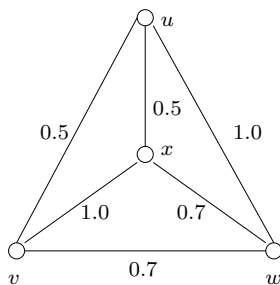


Figura 3.2: Exemplo para nós de corte e pontes fuzzy

Observação 3.4.2. (Sunitha e Vijayakumar, 1999) A condição do teorema acima não é necessária. Na Figura 3.2, w é um nó de corte fuzzy (Definição 3.4.2); (u, w) e (v, x) são as únicas pontes fuzzy.

Observação 3.4.3. (Sunitha e Vijayakumar, 1999) No seguinte grafo fuzzy (Figura 3.3), (u_1, u_2) e (u_3, u_4) são as pontes fuzzy e nenhum nó é um nó de corte fuzzy. Isto é uma diferença significativa da teoria de grafos clássica.

Teorema 3.4.4. (Sunitha e Vijayakumar, 1999) Se (u, v) é uma ponte fuzzy, então $\mu^\infty(u, v) = \mu(u, v)$.

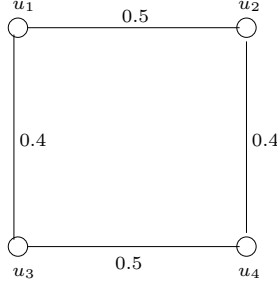


Figura 3.3: Exemplo de um grafo sem nós de corte fuzzy

Prova: Suponha que (u, v) é uma ponte fuzzy e que $\mu^\infty(u, v)$ excede $\mu(u, v)$. Então existe um caminho mais forte entre u e v com força maior que $\mu(u, v)$ e todos os arcos deste caminho mais forte têm força maior que $\mu(u, v)$. Agora, este caminho junto com o arco (u, v) formam um ciclo no qual (u, v) é o arco mais fraco, contradizendo o fato de (u, v) ser uma ponte fuzzy. \square

Observação 3.4.4. (Sunitha e Vijayakumar, 1999) *O inverso do teorema acima não é verdadeiro. A condição para o inverso ser verdadeiro é discutido no Teorema 3.5.6.*

G é chamado não-separável (um bloco) se ele não possui nós de corte. Mostraremos que um bloco pode ter pontes (isto não poderia acontecer para grafos não-fuzzy). Por exemplo na Figura 3.4

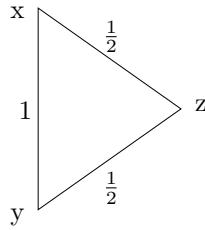


Figura 3.4: Exemplo de um grafo não separável

o arco (x, y) é uma ponte, já que sua retirada reduz a força de conexão entre x e y de 1 para $\frac{1}{2}$. Contudo, é fácil verificar que nenhum nó deste grafo fuzzy é um nó de corte.

Se entre todos os pares de nós x e y de G existem dois caminhos mais fortes que são disjuntos (exceto pelos próprios x, y), G é evidentemente um bloco. Isto é análogo ao “se” do teorema de grafos não fuzzy de que G é um bloco (com pelo menos três nós) se, e somente se, todos pares de nós de G pertencem a um ciclo comum. O “somente se”, por outro lado, não pertence ao caso fuzzy, como o exemplo dado mostra.

Teorema 3.4.5. (Bhattacharya, 1987) *Seja $G = (\sigma, \mu)$ um grafo fuzzy onde S é o conjunto de nós. Então, ou $\mu(x, y) = \mu(x, z) = \mu(z, y) = 0.5$ e $\sigma : S \rightarrow [0, 1]$ ou G tem pelo menos uma ponte.*

Prova: A primeira possibilidade é apresentada no seguinte exemplo. Seja $S = \{x, y, z\}$ e definimos $\mu(x, y) = \mu(x, z) = \mu(z, y) = 0.5$ e $\sigma : S \rightarrow [0, 1]$. Neste grafo fuzzy, como os arcos possuem mesmo peso, então removendo qualquer arco não há redução na força de conectividade entre quaisquer pares de nós.

Agora considere o caso em que $\mu(x, y)$ não é constante para todos os arcos de G . Escolha um arco (x_0, y_0) tal que $\mu(x_0, y_0)$ é um máximo no conjunto de todos os possíveis valores de $\mu(x, y)$ para $x, y \in S$. Então $\mu(x_0, y_0) > 0$ e existe pelo menos algum arco (u, v) distinto de (x_0, y_0) tal que $\mu(u, v) < \mu(x_0, y_0)$. Supomos que (x_0, y_0) seja uma ponte de G . Se removermos (x_0, y_0) então a força de conectividade entre x_0 e y_0 no subgrafo fuzzy obtido é decrescida. Em outras palavras

$$\mu'(x_0, y_0) < \mu(x_0, y_0).$$

Portanto, usando o Teorema 3.4.1, (x_0, y_0) é uma ponte de G . □

Corolário 3.4.1. (Bhattacharya, 1987) *Em um grafo fuzzy $G = (\sigma, \mu)$ para o qual μ não é um mapeamento constante, um arco (x, y) para o qual $\mu(x, y)$ é máximo, é uma ponte de G .*

Observação 3.4.5. (Bhattacharya, 1987) *O inverso do Corolário 3.4.1 não é verdade. Por exemplo, considere o grafo fuzzy mostrado na Figura 3.5. Aqui $\mu(x, y) = 0.75$ e $\mu'(x, y) = 0.5$.*

Então removendo o arco (x, y) reduzimos a força de conectividade entre os nós x e y . Portanto (x, y) é uma ponte mas vimos que $\mu(x, y)$ não é um máximo.

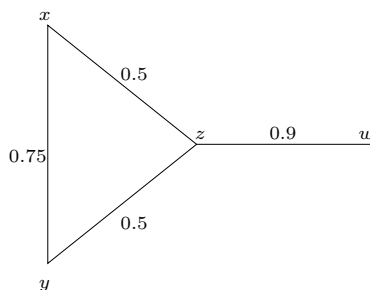


Figura 3.5: Um grafo em que $\mu(x, y)$ não é o máximo apesar de (x, y) ser uma ponte

Analisando o Teorema 3.4.5 é natural perguntar se todo grafo fuzzy tem pelo menos um nó de corte. A resposta é não como os exemplos a seguir mostram.

Exemplo 3.4.1. (Bhattacharya, 1987) Seja $S = \{x, y, z\}$, $\mu : S \times S \rightarrow [0, 1]$ definido por $\mu(x, y) = 1$, $\mu(y, z) = 0.5$, $\mu(x, z) = 0.5$ e μ é simétrico com $\sigma : S \rightarrow [0, 1]$. Então $G = (\sigma, \mu)$ não tem nós de corte.

Exemplo 3.4.2. (Bhattacharya, 1987) O exemplo 3.4.2 é um grafo fuzzy que é um “triângulo”. Então podemos perguntar se existe um grafo fuzzy com mais de três nós e que seja um bloco? Para construir um exemplo, seja $S = \{x, y, z, w, u\}$ e $\mu(x, y) = 1$, $\mu(y, u) = 0.5$; $\mu(u, x) = 1$, $\mu(x, w) = 0.5$, $\mu(w, z) = 1$, $\mu(z, x) = 0.5$ e seja μ simétrico e um a escolhido de forma arbitrária. É fácil checar que $\mu^2(u, v) = 1$, $\mu^3(u, v) = 1, \dots, \mu^\infty(u, v) = 1$. Contudo $\mu'^\infty(u, v) = 0.5$. Então usando o Teorema 3.4.1 temos que o arco (u, v) é uma ponte. É fácil verificar que este grafo fuzzy não possui nós de corte.

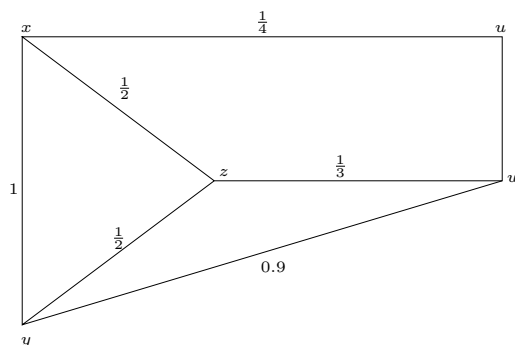


Figura 3.6: Grafo fuzzy com mais de um nó de corte

Exemplo 3.4.3. (Bhattacharya, 1987) Aqui será dado um exemplo de um grafo fuzzy com mais de um nó de corte. Considere o grafo fuzzy apresentado na Figura 3.6. Neste grafo fuzzy temos que $\mu^\infty(x, u) = 0.9$. Se o nó w é removido então a força de conectividade entre x e u é igual a 0.25 . Portanto w é um nó de corte. É fácil ver que y é um nó de corte mas z não é.

3.5 Florestas e Árvores

Lembramos que um grafo crisp que não possui ciclos é chamado de *acíclico* ou uma *floresta*. Uma floresta conectada é chamada de *árvore*. Chamamos um grafo fuzzy de floresta se o grafo consistindo de arcos não nulos é uma floresta, e uma árvore se este grafo é também conectado.

Definição 3.5.1. (Rosenfeld, 1975) Um grafo fuzzy $G = (\sigma, \mu)$ é chamado de floresta fuzzy se ele tem um subgrafo gerador fuzzy $F = (\sigma, \nu)$ que é uma floresta, onde para todos os arcos (x, y) que não estão em F (ou seja, tal que $\nu(x, y) = 0$), temos $\mu(x, y) < \nu^\infty(x, y)$. Em outras palavras, se $(x, y) \in G$ mas $\notin F$, existe um caminho em F entre x e y cuja força é maior que $\mu(x, y)$. Uma floresta é uma floresta fuzzy.

Por exemplo, os grafos da Figura 3.7 são florestas fuzzy

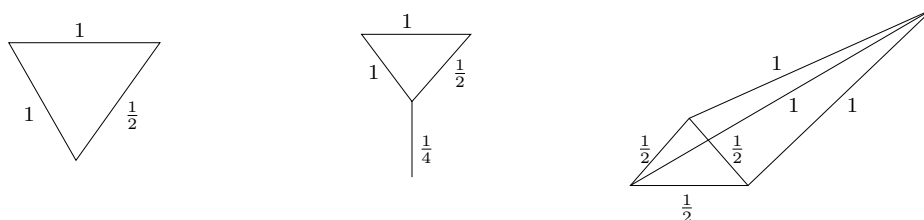


Figura 3.7: Exemplos de Floresta Fuzzy

pois a retirada dos arcos com grau de pertinência igual a $\frac{1}{2}$ de qualquer um dos grafos resulta em um subgrafo gerador fuzzy acíclicos e os caminhos entre os nós possuem força igual a 1. Contudo, os subgrafos da Figura 3.8 não são florestas fuzzy:

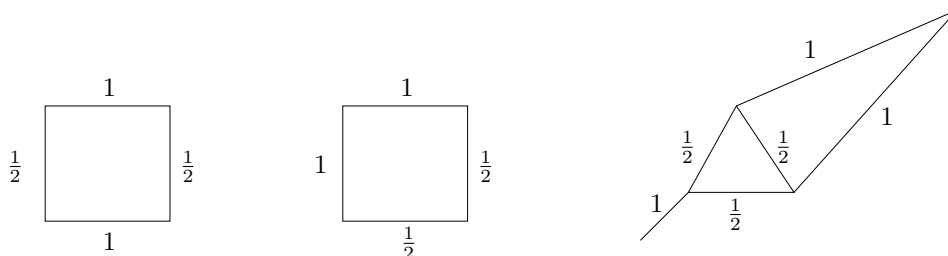


Figura 3.8: Exemplos de grafos que não são florestas

pois a retirada de um ou mais arcos com grau de pertinência igual a $\frac{1}{2}$ fornece um subgrafo gerador fuzzy cujos caminhos entre os nós é igual a $\frac{1}{2}$.

Se G é conectado, então F também é (todos os arcos de um caminho em G também estão em F , ou podem ser desviados para F). Neste caso, chamamos G uma *árvore fuzzy*. Os exemplos de florestas fuzzy dados acima são todos árvores fuzzy.

Note que se substituirmos $<$ por \leq na definição 3.5.1, então até o grafo da Figura 3.9 seria uma floresta fuzzy, já que possui subgrafos tais como mostrado na Figura 3.10.

Teorema 3.5.1. (Rosenfeld, 1975) G é uma floresta fuzzy se, e somente se, em qualquer ciclo de G existe um arco (x, y) tal que $\mu(x, y) < \mu'^\infty(x, y)$, onde μ' denota retirada do arco (x, y) de G .

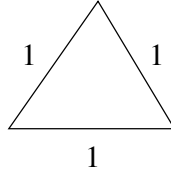


Figura 3.9: Exemplo de florestas fuzzy

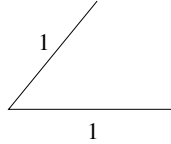


Figura 3.10: Subgrafos resultantes da figura anterior

Prova: Para ver “se”, seja (x, y) um arco que pertence a uma ciclo, possui a propriedade do teorema acima e para o qual $\mu(x, y)$ é o menor. (Se não há ciclos, G é uma floresta e esta parte já está demonstrada). Se retirarmos (x, y) , o subgrafo resultante satisfaz a propriedade de caminho de uma floresta fuzzy. Se ainda existirem ciclos neste grafo, podemos repetir o processo. Note que a cada estágio, nenhum arco previamente retirado é mais forte que o arco que deve ser retirado no momento. Portanto, o caminho garantido pela propriedade do teorema envolve somente arcos que ainda não foram retirados. Quando não houver mais ciclos, o subgrafo resultante é uma floresta F . Seja (x, y) um arco que não pertença a F . Então (x, y) é um dos arcos que foram eliminados no processo de construção de F , e existe um caminho de x para y que é mais forte que $\mu(x, y)$ e que não envolve (x, y) nem qualquer dos arcos retirados previamente. Se este caminho envolve arcos que foram retirados posteriormente, este pode ser desviado usando um caminho de arcos ainda mais fortes. Se o caminho ainda utilizar arcos que foram eliminados, existe um caminho mais forte, e assim por diante. Este processo eventualmente estabiliza com um caminho consistindo somente de arcos pertencentes a F . Então G é uma floresta fuzzy.

No outro sentido, se G é uma floresta fuzzy, seja ρ um ciclo qualquer. Então algum arco (x, y) de ρ não pertence a F . Portanto, pela definição de uma floresta fuzzy, temos $\mu(x, y) < \nu^\infty(x, y) \leq \mu'^\infty(x, y)$, o que provaria o “somente se”. \square

Note que se G é conectado, então F construída na primeira parte da prova também o será, já que em nenhum passo da construção haverá a desconexão.

Proposição 3.5.1. (Rosenfeld, 1975) *Se existe no máximo um caminho mais forte entre quaisquer dois nós de G , então G deve ser uma floresta fuzzy.*

Prova: Se não, pelo teorema 3.5.1, haveria um ciclo ρ em G tal que, para todos os arcos (x, y) de ρ , teríamos $\mu(x, y) > \mu'^{\infty}(x, y)$. Portanto, o próprio (x, y) constituiria um caminho mais forte de x para y . Se escolhermos (x, y) para ser o arco mais fraco de ρ segue que o resto de ρ é também um caminho mais forte de x para y , o que seria uma contradição. \square

O inverso da Proposição 3.5.1 é falso. G pode ser uma floresta fuzzy e ainda ter múltiplos caminhos mais fortes entre nós. Isto é devido à força de um caminho ser dada pelo arcos mais fracos, e enquanto este arco pertencer a F , existe pouca restrição sobre os outros arcos. Por exemplo, o grafo fuzzy da Figura 3.11

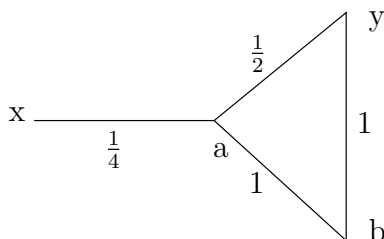


Figura 3.11: Exemplo de floresta fuzzy

é uma floresta fuzzy. Aqui F consiste de todos os arcos exceto (a, y) . Os caminhos mais fortes entre x e y têm força igual a $\frac{1}{4}$, devido ao arco (x, a) . Tanto x, a, b, y quanto x, a, y são caminhos, onde o primeiro pertence a F e o segundo não.

Proposição 3.5.2. (Rosenfeld, 1975) *Se G é uma floresta fuzzy, os arcos de F são pontes de G .*

Prova: Se um arco (x, y) não está em F certamente não é uma ponte, já que $\mu(x, y) < \nu^{\infty}(x, y) \leq \mu'^{\infty}(x, y)$. Por outro lado, seja (x, y) um arco pertencente a F . Se ele não for uma ponte, teríamos um caminho ρ de x para y , não envolvendo (x, y) de força $\geq \mu(x, y)$. Este caminho deve envolver arcos que não estão em F , já que F é uma floresta e não possui ciclos. Contudo, por definição, qualquer arco (u_i, v_i) pode ser substituído por um caminho ρ_i em F de força $> \mu(x, y)$. Agora, ρ_i não pode envolver (x, y) já que todos seus arcos são estritamente mais fortes que $\mu(x, y)$. Portanto substituindo cada (u_i, v_i) por ρ_i , podemos construir um caminho em F de x para y que não envolva (x, y) resultando em um ciclo em F , o que é uma contradição. \square

Por esta última proposição, se G é uma floresta fuzzy, sua floresta geradora F é única.

Teorema 3.5.2. (Sunitha e Vijayakumar, 1999) Se $G : (\sigma, \mu)$ é uma árvore fuzzy e $G^* : (\sigma^*, \mu^*)$ não é uma árvore, então existe pelo menos um arco (u, v) em μ^* para o qual $\mu(u, v) < \mu^\infty(u, v)$.

Prova: Se G é uma árvore fuzzy, então por definição existe um subgrafo gerador fuzzy $F : (\sigma, \nu)$, que é uma árvore e $\mu(u, v) < \mu^\infty(u, v)$ para todo arco (u, v) que não pertence a F . Também temos que $\nu^\infty(u, v) \leq \mu^\infty(u, v)$ pelo Lema 3.2.1. Portanto $\mu(u, v) < \mu^\infty(u, v)$ para todo (u, v) que não pertence a F e, por hipótese, existe pelo menos um arco (u, v) que não pertence a F , o que completa a prova. \square

Definição 3.5.2. (Bhutani, 1989). Um grafo completo fuzzy é um grafo fuzzy $G : (\sigma, \mu)$ tal que $\mu(u, v) = \sigma(u) \wedge \sigma(v)$ para todo u e v .

Lema 3.5.1. (Bhutani, 1989). Se G é um grafo fuzzy completo, então $\mu^\infty(u, v) = \mu(u, v)$.

Lema 3.5.2. (Bhutani, 1989). Um grafo completo fuzzy não possui nós de corte.

Observação 3.5.1. (Sunitha e Vijayakumar, 1999) O inverso do lema 3.5.1 não é verdadeiro (Figura 3.12a). Também, um grafo completo fuzzy pode ter uma ponte fuzzy (Figura 3.12b).

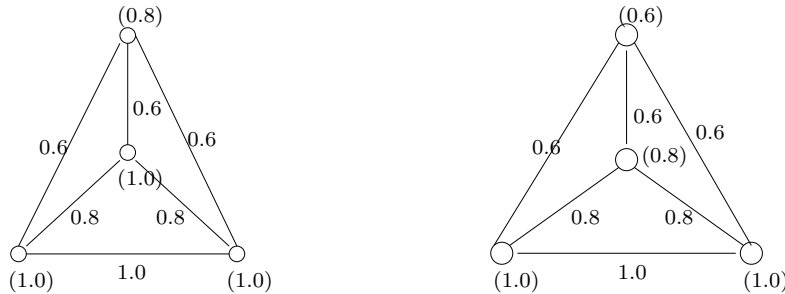


Figura 3.12: (a) e (b): Grafos completos fuzzy

Teorema 3.5.3. (Sunitha e Vijayakumar, 1999) Se $G : (\sigma, \mu)$ é uma árvore fuzzy, então G não é completo.

Prova: Seja G um grafo completo fuzzy. Então, $\mu^\infty(u, v) = \mu(u, v)$ para todo u, v (Lema 3.5.1). Sendo G uma árvore fuzzy, $\mu(u, v) < \nu^\infty(u, v)$ para todo $(u, v) \notin F$. Portanto, $\mu^\infty(u, v) < \nu^\infty(u, v)$, contradizendo o Lema 3.5.1. \square

Teorema 3.5.4. (Sunitha e Vijayakumar, 1999) *Se G é uma árvore fuzzy, então os arcos de F são as pontes fuzzy de G .*

Teorema 3.5.5. (Sunitha e Vijayakumar, 1999) *Se G é uma árvore fuzzy, então os nós internos de F são os nós de corte de G .*

Prova: Seja w qualquer nó de G que não seja um nó terminal de F . Então pelo Teorema 3.5.4, este é o nó comum a pelo menos dois arcos de F os quais são pontes fuzzy de G e, pelo Teorema 3.4.3, w é um nó de corte fuzzy. Também, se w é um nó terminal de F , então w não é um nó de corte fuzzy pois senão existiria um caminho entre u, v distintos de w para o qual w pertencesse a todos os caminhos mais fortes e um destes caminhos certamente está em F . Mas se w é um nó terminal de F , isto não seria possível. \square

Corolário 3.5.1. (Sunitha e Vijayakumar, 1999) *Um nó de corte fuzzy de uma árvore fuzzy é o nó comum de pelo menos duas pontes fuzzy.*

Teorema 3.5.6. (Sunitha e Vijayakumar, 1999) *$G : (\sigma, \mu)$ é uma árvore fuzzy se, e somente se, são equivalentes*

1. (u, v) é uma ponte fuzzy
2. $\mu^\infty(u, v) = \mu(u, v)$.

Prova: Seja $G : (\sigma, \mu)$ uma árvore fuzzy e seja (u, v) uma ponte fuzzy. Então $\mu^\infty(u, v) = \mu(u, v)$ (Teorema 3.4.4). Agora, seja (u, v) um arco em G tal que $\mu^\infty(u, v) = \mu(u, v)$. Se G^* é uma árvore, então claramente (u, v) é uma ponte fuzzy; caso contrário, segue do teorema 3.5.2 que $(u, v) \in F$ e (u, v) é uma ponte fuzzy (Teorema 3.5.4). \square

Observação 3.5.2. (Sunitha e Vijayakumar, 1999) *Para uma árvore fuzzy G , o subgrafo gerador F é único (Teorema 3.5.4). Segue da prova do teorema que F nada mais é do que a árvore geradora máxima T de G .*

Teorema 3.5.7. (Sunitha e Vijayakumar, 1999) Um grafo fuzzy é uma árvore fuzzy se, e somente se, ele tem uma única árvore geradora máxima.

Observação 3.5.3. (Sunitha e Vijayakumar, 1999) Para um grafo fuzzy que não é uma árvore fuzzy existe pelo menos um arco em T que não é uma ponte fuzzy e arcos que não estão em T e não são pontes fuzzy de G . Esta observação leva ao seguinte teorema.

Teorema 3.5.8. (Sunitha e Vijayakumar, 1999) Se $G : (\sigma, \mu)$ é um grafo fuzzy com $\sigma^* = S$ e $|S| = p$ então G tem no máximo $p - 1$ pontes fuzzy.

Teorema 3.5.9. (Sunitha e Vijayakumar, 1999) Seja $G : (\sigma, \mu)$ um grafo fuzzy e seja T a árvore geradora máxima de G . Então os nós terminais de T não são nós de corte fuzzy de G .

Corolário 3.5.2. (Sunitha e Vijayakumar, 1999) Todo grafo fuzzy tem pelo menos dois nós que não são nós de corte fuzzy.

Definição 3.5.3. (Nair e Cheng, 2001)

1. (σ, μ) é uma árvore se, e somente se, $(\text{supp}(\sigma), \text{supp}(\mu))$ é uma árvore.
2. (σ, μ) é uma árvore fuzzy se, e somente se, (σ, μ) tem um subgrafo gerador fuzzy (τ, ν) o qual é uma árvore tal que $\forall (u, v) \in \text{supp}(\mu) \setminus \text{supp}(\nu)$, $\mu(u, v) < \nu^\infty(u, v)$, isto é, existe uma caminho em (σ, ν) entre u e v cuja força é maior que $\mu(u, v)$.

Definição 3.5.4. (Nair e Cheng, 2001)

1. (σ, μ) é um ciclo se, e somente se, $(\text{supp}(\sigma), \text{supp}(\mu))$ é um ciclo.
2. (σ, μ) é um ciclo fuzzy se, e somente se, $(\text{supp}(\sigma), \text{supp}(\mu))$ é um ciclo e $\exists (x, y) \in \text{supp}(\mu)$ único tal que $\mu(x, y) = \bigwedge \{\mu(u, v) \mid (u, v) \in \text{supp}(\mu)\}$.

Exemplo 3.5.1. (Nair e Cheng, 2001) Sejam $S = \{u, v, w, s, t\}$ e $X = \{(u, v), (u, w), (v, w), (w, s), (w, t), (s, t)\}$. Seja $\sigma(x) = 0.9$ para todo $x \in S$ e seja μ o subconjunto fuzzy de X definido por $\mu(u, v) = 0.4$, $\mu(u, w) = \mu(v, w) = \mu(w, s) = \mu(w, t) = \mu(s, t) = 0.9$. Então (σ, μ) não é nem um ciclo fuzzy, nem uma árvore fuzzy (Figura 3.13).

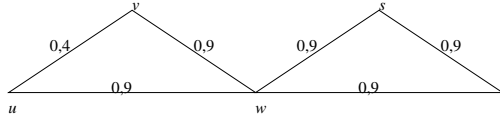


Figura 3.13: Figura do exemplo 3.5.1

Exemplo 3.5.2. (Nair e Cheng, 2001) Seja $S = \{w, u, v\}$ e $X = \{(w, u), (w, v), (u, v)\}$. Seja $\sigma(x) = 0.8$ para todo $x \in S$ e μ e μ' subconjuntos fuzzy de X definido por $\mu(w, u) = \mu(w, v) = 0.8$, $\mu(u, v) = 0.2$ e $\mu'(w, v) = 0.8$, $\mu'(w, u) = \mu'(u, v) = 0.2$. Então (σ, μ) é uma árvore fuzzy, mas não é uma árvore e não é um ciclo fuzzy enquanto (σ, μ') é um ciclo fuzzy, mas não uma árvore fuzzy.

Teorema 3.5.10. (Nair e Cheng, 2001) Seja (σ, μ) um ciclo. Então (σ, μ) é um ciclo fuzzy se, e somente se, (σ, μ) não é uma árvore fuzzy.

Teorema 3.5.11. (Nair e Cheng, 2001) Se $\exists \alpha \in (0, 1]$ tal que $(\text{supp}(\sigma), \mu^\alpha)$ é uma árvore, μ^α um α -corte, então (σ, μ) é uma árvore fuzzy. Inversamente, se (σ, μ) é um ciclo e (σ, μ) é uma árvore fuzzy, então $\exists \alpha \in (0, 1]$ tal que $(\text{supp}(\sigma), \mu^\alpha)$ é uma árvore.

Ilustrando o Teorema 3.5.11

Exemplo 3.5.3. (Nair e Cheng, 2001) Seja $S = \{s, t, u, v, w\}$ e $X = \{(s, t), (s, u), (t, u), (u, v), (u, w), (w, v)\}$. Seja $\sigma(x) = 0.7$ para todo $x \in S$ e seja μ um subconjunto fuzzy de X definido por $\mu(s, t) = 0.4$, $\mu(s, u) = \mu(t, u) = 0.5$, $\mu(u, v) = 0.6$ e $\mu(u, w) = \mu(w, v) = 0.7$. Então $\exists q \in (0, 1]$ tal que $(\text{supp}(\sigma), \mu^q)$ é uma árvore. Contudo (σ, μ) é uma árvore fuzzy.

Exemplo 3.5.4. (Moderson e Nair, 1996a) Sejam $V = \{w, u, v\}$ e $X = \{(w, u), (w, v), (u, v)\}$. Seja $\sigma = \delta_V$ e μ e μ' subconjuntos fuzzy de X definido por $\mu(w, u) = \mu(w, v) = 1$, $\mu(u, v) = \frac{1}{2}$ e $\mu'(w, u) = 1$, $\mu'(w, v) = \mu'(u, v) = \frac{1}{2}$. Então $G = (\sigma, \mu)$ é uma árvore fuzzy, mas não é uma árvore e não é um ciclo fuzzy, ao passo que (σ, μ') é um ciclo fuzzy, mas não uma árvore fuzzy.

Teorema 3.5.12. (Moderson e Nair, 1996a) Seja $G = (\sigma, \mu)$ um ciclo. Então (σ, μ) é um ciclo fuzzy se, e somente se, $G = (\sigma, \mu)$ não é uma árvore fuzzy.

Prova: Suponha que $G = (\sigma, \mu)$ é um ciclo fuzzy. Então \exists arcos distintos $x_1, x_2 \in \mu^*$ tal que $\mu(x_1) = \mu(x_2) = \min\{\mu(y) | y \in \mu^*\}$. Se (σ, ν) é qualquer árvore geradora de $G = (\sigma, \mu)$

(Rosenfeld, 1975), então $\mu^* - \nu^* = \{(u, v)\}$ para algum $u, v \in V$ desde que $G = (\sigma, \mu)$ é um ciclo. Portanto $G = (\sigma, \mu)$ não é uma árvore fuzzy. Inversamente, suponha que $G = (\sigma, \mu)$ não é uma árvore fuzzy. Como $G = (\sigma, \mu)$ é um ciclo, temos que $\forall (u, v) \in \mu^*$ que (σ, ν) é um subgrafo gerador fuzzy de $G = (\sigma, \mu)$, que é uma árvore, e $\nu^\infty(u, v) \leq \mu(u, v)$, onde $\nu(u, v) = 0$ e $\nu(y) = \mu(y) \forall y \in \mu^* \setminus \{(u, v)\}$. Portanto μ não atinge $\min\{\mu(y) | y \in \mu^*\}$ unicamente. Portanto $G = (\sigma, \mu)$ é um ciclo fuzzy. \square

Teorema 3.5.13. (Moderson e Nair, 1996a) *Se $\exists q \in (0, 1]$ tal que (σ^*, μ_q) é uma árvore, então $G = (\sigma, \mu)$ é uma árvore fuzzy. Inversamente, se $G = (\sigma, \mu)$ é um ciclo e $G = (\sigma, \mu)$ é uma árvore fuzzy, então $\exists q \in (0, 1]$ tal que (σ^*, μ_q) é uma árvore.*

Prova: Suponha que q exista. Seja ν o subconjunto fuzzy de X tal que $\nu = \mu$ sobre μ_q e $\nu(x) = 0$ se $x \in X \setminus \mu_q$. Então $H = (\sigma, \nu)$ é um subgrafo gerador fuzzy de $G = (\sigma, \mu)$ tal que (σ, ν) é uma árvore fuzzy já que (σ^*, ν^*) é uma árvore. Suponha que $(u, v) \in X$ e $(u, v) \notin \nu_q$. Então \exists um caminho entre u e v de força $\geq q > \mu(u, v)$. Então $G = (\sigma, \mu)$ é uma árvore fuzzy. Pelo caminho inverso, notamos que desde que $G = (\sigma, \mu)$ é um ciclo e uma árvore fuzzy, $\exists x \in \mu^*$ único tal que $\mu(x) = \min\{\mu(y) | y \in \mu^*\}$. Seja q tal que $\mu(x) < q \leq \min\{\mu(y) | y \in \mu^* \setminus \{x\}\}$. Então (σ^*, μ_q) é uma árvore. \square

Exemplo 3.5.5. (Moderson e Nair, 1996a) *Seja $V = \{s, t, u, v, w\}$ e $X = \{(s, t), (s, u), (t, u), (u, v), (u, w), (w, v)\}$. Seja $\sigma = \delta_V$ e seja μ subconjuntos fuzzy de X definidos por $\mu(s, t) = \frac{1}{4}, \mu(s, u) = \mu(t, u) = \frac{3}{8}, \mu(u, v) = \frac{1}{2},$ e $\mu(u, w) = \mu(w, v) = 1$. Então $\nexists q \in (0, 1]$ tal que (σ^*, μ_q) é uma árvore. Contudo $G = (\sigma, \mu)$ é uma árvore fuzzy.*

Definição 3.5.5. (Moderson e Nair, 1996a) *Seja E um subconjunto de μ^* .*

1. $\{x_{\mu(x)} | x \in E\}$ é um conjunto de corte de $G = (\sigma, \mu)$ se, e somente se, E é um conjunto de corte de (σ^*, μ^*) .
2. $\{x_{\mu(x)} | x \in E\}$ é um conjunto de corte fuzzy de $G = (\sigma, \mu)$ se, e somente se, $\exists u, v \in \sigma^*$ tal que $\mu'^\infty(u, v) < \mu(u, v)$, onde μ' é o subconjunto fuzzy de X definido por $\mu' = \mu$ sobre $\mu^* \setminus E$ e $\mu'(x) = 0 \forall x \in E$, isto é, a remoção de E de μ^* reduz a força de conectividade entre alguns pares de nós de $G = (\sigma, \mu)$.

Quando E é um conjunto *singleton*, um conjunto de corte é chamado de *ponte* e um conjunto de corte fuzzy é chamado de uma *ponte fuzzy*.

Nossa definição de ponte fuzzy coincide com a definição de ponte em (Rosenfeld, 1975). O seguinte exemplo é um grafo fuzzy $G = (\sigma, \mu)$ que não possui pontes fuzzy e onde μ não é uma função constante.

Exemplo 3.5.6. (Moderson e Nair, 1996a) Sejam $V = \{t, u, v, w\}$ e $X = \{(t, u), (u, v), (v, w), (w, t), (t, v)\}$. Seja $\sigma = \delta_V$, $\mu(u, v) = \mu(u, v) = \mu(v, w) = \mu(w, t) = 1$ e $\mu(t, v) = \frac{1}{2}$. Então μ não é constante, mas $G = (\sigma, \mu)$ não possui uma ponte fuzzy uma vez que a força de conectividade entre quaisquer pares de nós de $G = (\sigma, \mu)$ resulta em 1, mesmo após a remoção de um arco.

Teorema 3.5.14. (Moderson e Nair, 1996a) Sejam $V = \{v_1, \dots, v_n\}$ e $C = \{v_1v_2, v_2v_3, \dots, v_{n-1}v_n, v_nv_1\}$, $n \geq 3$.

1. Suponha que $\mu^* \supseteq C$ e que $\forall v_jv_k \in \mu^* - C$, $\mu(v_j, v_k) < \max\{\mu(v_i v_{i+1}) \mid i = 1, \dots, n\}$, onde $v_{n+1} = v_1$. Então ou μ é uma constante sobre C ou $G = (\sigma, \mu)$ é uma ponte fuzzy.
2. Suponha que $0 \neq \mu^* \subset C$. Então (σ, μ) têm uma ponte.

Prova:

1. Suponha que μ não é constante sobre C . Seja $v_h, v_{h+1} \in C$ tal que $\mu(v_h, v_{h+1}) = \max\{\mu(v_i, v_{i+1}) \mid i = 1, \dots, n\}$. Já que μ não é constante sobre C , a força do caminho $C - \{v_h v_{h+1}\}$ ente v_h e v_{h+1} é estritamente menor que $\mu(v_h, v_{h+1})$. A força de qualquer outro caminho P entre v_h e v_{h+1} é também estritamente menor que $\mu(v_h, v_{h+1})$ já que P deve conter um arco de $\mu^* \setminus C$. Portanto $(v_h, v_{h+1})_{\mu(v_h, v_{h+1})}$ é uma ponte fuzzy.
2. Imediato.

□

Teorema 3.5.15. (Moderson e Nair, 1996a) Suponha que a dimensão do espaço do ciclo de (σ^*, μ^*) seja 1 (Harary, 1972). Então $G = (\sigma, \mu)$ não tem uma ponte fuzzy se, e somente se, $G = (\sigma, \mu)$ é um ciclo e μ é uma função constante.

Prova: Suponha que este não é o caso de $G = (\sigma, \mu)$ é um ciclo e μ é uma função constante. Se $G = (\sigma, \mu)$ não é um ciclo, então \exists um arco $x \in \mu^*$ que não é parte do ciclo. Então $x_{\mu(x)}$ é uma ponte e portanto uma ponte fuzzy. Suponha que $G = (\sigma, \mu)$ é um ciclo, mas μ não é uma função constante. Seja $x \in \mu^*$ tal que $\mu(x)$ é maximal. Então x_μ é uma ponte fuzzy. Inversamente, suponha que $G = (\sigma, \mu)$ é um ciclo e μ é uma função constante. Então a remoção de um arco $v_i v_{i+1}$ resulta em um caminho único ente v_i e v_{i+1} de força igual a $\mu(v_i v_{i+1})$. Portanto $(v_i v_{i+1})_{\mu(v_i v_{i+1})}$ não é uma ponte fuzzy. □

3.6 Cliques Fuzzy

Definição 3.6.1. (Nair e Cheng, 2001) Seja G um grafo fuzzy sobre S e $H = (\tau, \nu)$ um subgrafo induzido por $T \subseteq S$. Então H é um clique se $(\text{supp}(\tau), \text{supp}(\nu))$ é um clique e H é um clique fuzzy se H é um clique e todo ciclo em (τ, ν) é um ciclo fuzzy.

Exemplo 3.6.1. Seja $S = \{u, v, w\}$ e $X = \{(u, v), (u, w), (w, v)\}$. Seja $\sigma(x) = 1$ para todo $x \in S$ e seja ρ um subconjunto fuzzy de X definido por $\mu(u, v) = 0.7$, $\mu(u, w) = 0.5$ e $\mu(w, v) = 0.8$. Seja T o mesmo que S . Portanto $H = (\tau, \nu)$ é igual a $G = (\sigma, \mu)$. Claramente, u, v, w é um ciclo; mas não um ciclo fuzzy. Portanto H é um clique, mas não um clique fuzzy.

Exemplo 3.6.2. Seja $S = \{u, v, w\}$ e $X = \{(u, v), (u, w), (w, v)\}$. Seja $\sigma(x) = 1$ para todo $x \in S$ e seja μ o subconjunto fuzzy de X definido por $\mu(u, v) = \mu(u, w) = 0.5$ e $\mu(w, v) = 0.8$. Seja T igual a S . Então $H = (\tau, \nu)$ é igual a $G = (\sigma, \mu)$. Claramente, u, v, w é um ciclo e também é um ciclo fuzzy. Portanto H é um clique e um clique fuzzy.

Exemplo 3.6.3. Seja $S = \{s, u, v, w\}$ e $X = \{(s, u), (s, v), (s, w), (u, v), (u, w), (w, v)\}$. Seja $\sigma(x) = 1$ para todo $s \in S$ e seja μ um subconjunto fuzzy de X definido por $\mu(s, u) = 0.5$, $\mu(s, v) = \mu(s, w) = 0.8$, $\mu(u, v) = 0.6$, $\mu(u, w) = 0.8$ e $\mu(v, w) = 0.5$. Seja T igual a S . Portanto $H = (\tau, \nu)$ é o mesmo que $G = (\sigma, \mu)$. Claramente, s, u, v, w é um ciclo, mas não um ciclo fuzzy. Portanto H é um clique, mas não um clique fuzzy.

Exemplo 3.6.4. Seja $S = \{s, u, v, w\}$ e $X = \{(s, u), (s, v), (s, w), (u, v), (u, w), (w, v)\}$. Seja $\sigma(x) = 1$ para todo $s \in S$ e seja μ um subconjunto fuzzy de X definido por $\mu(s, u) = 0.5$, $\mu(s, v) = 0.8$, $\mu(s, w) = 0.7$, $\mu(u, v) = 0.5$, $\mu(u, w) = 0.5$ e $\mu(v, w) = 0.7$. Seja T o mesmo que S . Portanto $H = (\tau, \nu)$ é o mesmo que $G = (\sigma, \mu)$. Claramente todo ciclo é um ciclo fuzzy. Portanto H é um clique e um clique fuzzy.

Teorema 3.6.1. (Nair e Cheng, 2001) Seja G um grafo fuzzy sobre S e $H = (\tau, \nu)$ um subgrafo induzido por $T \subseteq S$. Então H é um clique fuzzy se, e somente se, todo ciclo de comprimento 3 em H é um ciclo fuzzy.

Prova: Assuma que H é um clique fuzzy. Por definição todo ciclo é um ciclo fuzzy e com isso todo ciclo de comprimento 3 é também um ciclo fuzzy.

Assuma que todo ciclo de comprimento 3 é um ciclo fuzzy. Vamos provar agora, por indução, que todo ciclo de tamanho ≥ 3 é um ciclo fuzzy. Todo ciclo de comprimento 3 é um ciclo fuzzy por suposição. A hipótese de indução é que todo ciclo de tamanho n é um ciclo fuzzy. Seja $x_0, x_1, \dots, x_n, x_{n+1}$ qualquer ciclo de comprimento $n+1$ em H . Como H é um clique, x_0, x_1, \dots, x_n é um ciclo de tamanho n em H e com isso é um ciclo fuzzy em H . Por isso existe pelo menos dois arcos, ditos e_1 e e_2 no ciclo fuzzy x_0, x_1, \dots, x_n tal que $\mu(e_1) = \mu(e_2) = \bigwedge \{\mu(e) | e \text{ é um arco em } x_0, x_1, \dots, x_n\}$. Agora, x_n, x_{n+1}, x_0 é um ciclo fuzzy e portanto existe pelo menos dois arcos, ditos e_3 e e_4 no ciclo fuzzy x_n, x_{n+1}, x_0 tal que $\mu(e_3) = \mu(e_4) = \bigwedge \{\mu(e) | e \text{ é um arco em } x_n, x_{n+1}, x_0\}$. Primeiro, consideramos o caso de um dos arcos e_1 ou e_2 é igual a um dos arcos e_3 e e_4 . Neste caso, sem perda de generalidade, assume-se que $e_1 = e_3$. Então e_2 e e_4 são arcos em $x_0, x_1, \dots, x_n, x_{n+1}$ e $\mu(e_2) = \mu(e_4) = \bigwedge \{\mu(e) | e \text{ é um arco em } x_0, x_1, \dots, x_n, x_{n+1}\}$. Portanto o resultado desejado. Se este não for o caso, note que todos os quatro arcos e_1, e_2, e_3, e_4 são arcos em $x_0, x_1, \dots, x_n, x_{n+1}$ e $\mu(e_1) = \mu(e_2) = \bigwedge \{\mu(e) | e \text{ é um arco em } x_0, x_1, \dots, x_n, x_{n+1}\}$ ou $\mu(e_3) = \mu(e_4) = \bigwedge \{\mu(e) | e \text{ é um arco em } x_0, x_1, \dots, x_n, x_{n+1}\}$ é verdade. \square

Lema 3.6.1. (Nair e Cheng, 2001) *Seja G um grafo fuzzy sobre S e $H = (\tau, \nu)$ um subgrafo induzido por $T \subseteq S$. Então todo ciclo de comprimento 3 em H é um ciclo fuzzy se, e somente se, para quaisquer três vértices, x, y, z em H tal que os arcos (x, y) e (y, z) estão em H_t implica (x, z) está em H_t para todo $0 \leq t \leq 1$.*

Definição 3.6.2. (Nair e Cheng, 2001) *Um subgrafo H de um grafo G é chamado de uma união disjunta de cliques se $V(H)$, o conjunto de nós de H , pode ser particionado em H_1, H_2, \dots, H_k tal que $\forall x, y \in V(H)$, (x, y) é um arco em H se, e somente se, $\{x, y\} \subseteq H_i$ para algum i , $1 \leq i \leq k$.*

Lema 3.6.2. (Nair e Cheng, 2001) *Seja G um grafo fuzzy sobre S e $H = (\tau, \nu)$ um subgrafo induzido por $T \subseteq S$. Então H é uma união disjunta de cliques se, e somente se, para quaisquer três nós x, y, z em H tal que os arcos (x, y) e (y, z) estão em H implica que (x, z) está em H .*

Prova: Seja H uma união disjunta de cliques e seja a partição de nós correspondentes H_1, H_2, \dots, H_k . Seja x, y, z qualquer três nós em H tal que os arcos (x, y) e (y, z) estão em H . Então $\{x, y\} \subseteq H_i$ e $\{y, z\} \subseteq H_j$ para $1 \leq i, j \leq k$. Já que $y \in H_i \cap H_j$, H_i é o mesmo que H_j . Portanto $\{x, z\} \subseteq H_j$. \square

Inversamente assume-se que H é um subgrafo de um grafo G tal que dados quaisquer três nós x, y, z em H , se arcos (x, y) e (y, z) são arcos em H então (x, z) é um arco de H . Seja x_1, x_2, \dots, x_n um caminho maximal em H . Alega-se que x_1, x_2, \dots, x_n é um clique em H . Para provar nossa alegação, tudo o que temos que provar é (x_i, x_j) é um arco de H , onde $1 \leq i < j + 1 \leq n$. Uma vez que (x_i, x_{i+1}) e (x_{i+1}, x_{i+2}) são arcos em H , (x_i, x_{i+2}) é um arco em H . Se $j = i + 2$, já está provado. Caso contrário, usamos o fato que (x_i, x_{i+2}) e (x_{i+2}, x_{i+3}) são arcos de H para concluir que (x_i, x_{i+3}) é um arco de H . Se $j = i + 3$, já está demonstrado. Caso contrário, repetindo o uso do argumento acima podemos mostrar que (x_i, x_{i+p}) é um arco em H e $j = i + p$.

Teorema 3.6.2. (Nair e Cheng, 2001) *Seja G um grafo fuzzy sobre S e $H = (\tau, \nu)$ um subgrafo induzido por $T \subseteq S$. Então H é um clique fuzzy se, e somente se, todo conjunto de corte de H é uma união disjunta de cliques.*

Prova: H é um clique fuzzy \Leftrightarrow todo ciclo de comprimento 3 em H é um ciclo fuzzy (pelo Teorema 3.6.1) \Leftrightarrow para quaisquer três nós x, y, z em H tal que os arcos (x, y) e (y, z) estão em H_t implica (x, z) está em H_t , $0 \leq t \leq 1$ (pelo Lema 3.6.1) \Leftrightarrow todo conjunto de corte de H é uma união disjunta de cliques (pelo Lema 3.6.2). \square

3.7 Ciclos fuzzy e cociclos fuzzy

Assumimos durante toda esta seção que (σ^*, μ^*) é conectado e que $\mu^* = X$ (Harary, 1972).

Definição 3.7.1. (Moderson e Nair, 1996a) *Seja (σ, ν) um subgrafo gerador fuzzy de (σ, μ) que é uma árvore. Se (σ, ν_f) é uma árvore fuzzy que $\nu \subseteq \nu_f \subseteq \mu$, $\nu_f = \mu$ sobre ν_f^* , e \bar{A} uma árvore fuzzy (σ, ν') tal que $\nu_f \subset \nu' \subseteq \mu$ e $\nu' = \mu$ sobre ν'^* , então (σ, ν_f) é chamada de árvore geradora fuzzy de (σ, μ) com respeito a ν .*

Claramente dado (σ, ν) e (σ, μ) da Definição 3.1.1, (σ, ν_f) existe. Nota-se que (σ, ν) é uma subgrafo gerador fuzzy de (σ, ν_f) .

No Exemplo 3.5.6, seja ν, ν_f e ν'_f subconjuntos fuzzy de X definidos a seguir: $\nu = \mu$ sobre $\{(t, u), (t, v), (t, w)\}$ e $\nu(w, v) = \nu(u, v) = 0$, $\nu_f = \mu$ sobre $\{(t, u), (t, v), (t, w), (u, v)\}$ e $\nu(w, v) = 0$, $\nu_f = \mu$ sobre $\{(t, u), (t, v), (t, w), (w, v)\}$ e $\nu'_f(u, v) = 0$. Então ambos (σ, ν_f) e (σ, ν'_f) são

árvores geradoras de (σ, μ) com respeito a ν . Isto é, dado ν , ν_f na Definição 3.7.1 não é necessariamente única.

Considere o grafo fuzzy (σ, μ) do Exemplo 3.5.4. Defina o subconjunto fuzzy ν de X por $\nu(w, u) = \nu(w, v) = 1$. Desde que $G = (\sigma, \mu)$ não é um ciclo fuzzy, a adição de $(u, v)_{\frac{1}{2}}$ para (σ, ν) não cria um ciclo fuzzy. Este fato motiva a seguinte definição de um *chord* fuzzy.

Definição 3.7.2. (Moderson e Nair, 1996a) *Seja (σ, ν) um subgrafo gerador fuzzy de $G = (\sigma, \mu)$, o qual é uma árvore. Seja $x \in \mu^*$.*

1. $x_{\mu(x)}$ é um chord de (σ, ν) se, e somente se, $x \notin \nu^*$, isto é, x é um chord de (σ^*, μ^*) (Harary, 1972).
2. $x_{\mu(x)}$ é um chord fuzzy de (σ, ν_f) se, e somente se, $x \notin \nu_f^*$.

Exemplo 3.7.1. (Moderson e Nair, 1996a) *Seja $V = \{s, t, u, v, w\}$ e $X = \{(w, s), (w, t), (w, u), (w, v), (s, t), (u, v)\}$. Definimos os subconjuntos fuzzy σ de V e μ, ν de X por $\sigma = \delta_V$ e $\mu = \delta_X$ sobre $X - \{(w, u)\}$ e $\mu(w, u) = \frac{1}{2}$, $\nu = \delta_Y$, onde $Y = X - \{(w, s), (w, u)\}$. Então $\nu_f = \mu$ sobre $X - \{(w, s)\}$ e $\nu_f(w, s) = 0$. Também $(w, s)_1$ e $(w, u)_{\frac{1}{2}}$ são chords de (σ, ν) e $(w, s)_1$ é um chord fuzzy de (σ, ν_f) . Se definirmos o subconjunto fuzzy ν' de X por $\nu' = \nu_f$ sobre ν_f^* e $\nu'(w, s) = t$ onde $0 < t < 1$, então (σ, ν') é uma árvore fuzzy tal que $\nu_f \subset \nu'$. Contudo, $\nu' \neq \mu$ sobre ν'^* .*

Definição 3.7.3. (Moderson e Nair, 1996a) *Seja (σ, ν) um subgrafo gerador fuzzy de (σ, μ) , que é uma árvore*

1. *Seja ν' um subconjunto fuzzy de X . Então (σ, ν') é uma co-árvore de (σ, ν) se, e somente se, $\forall x \in \mu^*, \nu'(x) = 0$ se $\nu(x) > 0$ e $\nu'(x) = \mu(x)$ se $\nu(x) = 0$.*
2. *Seja ν'_f um subconjunto fuzzy de X . Então (σ, ν'_f) é a co-árvore de (σ, ν_f) se, e somente se, $\forall x \in \mu^*, \nu'_f(x) = 0$ se $\nu_f(x) > 0$ e $\nu'_f(x) = \mu(x)$ se $\nu_f(x) = 0$.*

Seja (σ, ν') uma co-árvore de (σ, ν) , onde (σ, ν) é um subgrafo gerador fuzzy de (σ, μ) , que é uma árvore. Então (σ^*, ν'^*) é uma co-árvore de (σ^*, ν^*) já que $\nu'^* \cap \nu^* = 0, \nu'^* \cup \nu^* = \mu^*$ e (σ^*, ν^*) é uma árvore.

Definição 3.7.4. (Moderson e Nair, 1996a) *Seja (σ, ν) um subgrafo gerador fuzzy de (σ, μ) , que é uma árvore e seja $x \in \mu^*$.*

1. *Seja (σ, ν') uma co-árvore de (σ, ν) . Então $x_{\mu(x)}$ é um twig de (σ, ν') se, e somente se, $\nu'(x) = 0$.*

2. Seja (σ, ν'_f) uma co-árvore de (σ, ν_f) . Então $x_{\mu(x)}$ é um twig fuzzy de (σ, ν'_f) se, e somente se, $\nu'_f(x) = 0$.

Exemplo 3.7.2. (Moderson e Nair, 1996a) Seja (σ, μ) , (σ, ν) e (σ, ν_f) subgrafos fuzzy do Exemplo 3.7.1. Seja (σ, ν') a co-árvore de (σ, ν) e seja (σ, ν'_f) a coárvore de (σ, ν_f) . Então os twigs de (σ, ν') são $(s, t)_1, (w, t)_1, (w, v)_1$ e $(u, v)_1$. Os twigs fuzzy de (σ, ν'_f) são $(s, t)_1, (w, t)_1, (w, v)_1, (u, v)_1$ e $(w, u)_{\frac{1}{2}}$.

Definição 3.7.5. (Moderson e Nair, 1996a) Seja $x \in X$. Então x é chamado excepcional em G se, e somente se, \exists um ciclo $C \subseteq X$ tal que $x \in C$ e x é único com respeito a $\mu(x) = \min\{\mu(y)|y \in C\}$. Seja $E = \{x \in X|x \text{ é excepcional}\}$. Chamamos μ_E um subconjunto fuzzy de X definido por $\mu_E = \mu$ sobre $X - E$ e $\mu_E(x) = 0 \forall x \in E$.

Seja $S - \mu = \{x_t|x \in \mu^*, t \in (0, 1]\}$. Seja a adição de elementos de μ^* módulo 2 (Harary, 1972). Então $\forall x_t, y_s \in S_\mu, x_t + y_s = (x + y)_r$, onde $r = \min\{t, s\}$. Claramente $(S_\mu, +)$ é um semigrupo comunicativo com identidade 0_1 . Se S é um conjunto de singletons fuzzy, definimos $foot(S) = \{x|x_t \in S\}$.

Temos que $\sum \epsilon_i(x_i)_{\mu(x_i)} + \sum \epsilon'_i(x_i)_{\mu(x_i)} = \sum(\epsilon_i + \epsilon'_i)(x_i)_{\mu(x_i)}$, $\epsilon_i(x_i)_{\mu(x_i)} = (x_i)_{\mu(x_i)}$ se $\epsilon_i = 1$ e $\epsilon(x_i)_{\mu(x_i)} = (x_i) = 0$ se $\epsilon_i = 0$, $\epsilon_i, \epsilon'_i \in \mathcal{Z}_E$, . Lembramos que $\sum \epsilon_i(x_i)_{\mu(x_i)} = (\sum \epsilon_i x_i)_m$, onde $m = \min_i\{\mu(x_i)\}$.

Definição 3.7.6. (Moderson e Nair, 1996a)

1. $\sum \epsilon_i(x_i)_{\mu(x_i)}$ é um 1-chain com fronteira 0 em (σ, μ) , onde $x_i \in \mu^*$ se, e somente se, $\mu \epsilon_i x_i$ é uma 1-chain com fronteira 0 em (σ^*, μ^*) ((Harary, 1972),p.37).
2. $\sum \epsilon_i(x_i)_{\mu(x_i)}$ é um 1-chain com fronteira 0 fuzzy em (σ, μ) , onde $(x_i \in \mu_E^*)$ se, e somente se, $\sum \epsilon_i x_i$ é um 1-chain com fronteira 0 em (σ^*, μ_E^*) .

Um 1-chain com fronteira 0 (fuzzy) em (σ, μ) é chamado um cycle vector fuzzy.

Definição 3.7.7. (Moderson e Nair, 1996a)

1. $\sum \epsilon_i(x_i)_{\mu(x_i)}$ é um co-fronteira de (σ, μ) onde $x_i \in \mu^*$, se, e somente se, $\sum \epsilon_i x_i$ é uma co-fronteira de (σ^*, μ^*) .
2. $\sum \epsilon_i(x_i)_{\mu(x_i)}$ é uma co-fronteira de (σ, μ) , onde $x_i \in \mu_E^*$ se, e somente se, $\sum \epsilon_i x_i$ é uma co-fronteira de (σ^*, μ_E^*) .

$S' \subseteq S_\mu$ é chamado um cociclo (fuzzy) de (σ, μ) se, e somente se, $foot(S')$ é um cociclo de $((\sigma^*, \mu_E^*))(\sigma^*, \mu^*)$.

Definição 3.7.8. (Moderson e Nair, 1996a)

1. O conjunto de todos os cycle vectors (fuzzy) de (σ, μ) é chamado de conjunto ciclo (fuzzy) de (σ, μ) .
2. O conjunto de todas as co-fronteiras (fuzzy) de (σ, μ) é chamado de cociclos (fuzzy) de (σ, μ) .

Os seguintes exemplos mostram que o ciclo fuzzy, ciclo, cociclo fuzzy e conjunto de cociclos não são e não necessariamente geram espaços vetoriais sobre \mathcal{Z}_2 .

Exemplo 3.7.3. (Moderson e Nair, 1996a) Seja $V = \{t, u, v, w\}$ e $X = \{(t, u), (u, v), (v, w), (w, t), (t, v)\}$.

Definimos os subconjuntos fuzzy σ de V e μ de X a seguir: $\sigma = \delta_V$, $\mu(t, u) = \mu(u, v) = 1$, $\mu(v, w) = \mu(w, t) = \frac{1}{2}$ e $\mu(t, v) = \frac{1}{4}$. Então o conjunto de ciclos é $\{(t, u)_1 + (u, v)_1 + (t, v)_{\frac{1}{4}}, (v, w)_{\frac{1}{2}} + (w, t)_{\frac{1}{2}} + (t, v)_{\frac{1}{4}}, (t, u)_1 + (u, v)_1 + (v, w)_{\frac{1}{2}} + (w, t)_{\frac{1}{2}}, 0_{\frac{1}{4}}, 0_{\frac{1}{2}}\}$. O conjunto de ciclos fuzzy é $\{(t, u)_1 + (u, v)_1 + (v, w)_{\frac{1}{2}} + (w, t)_{\frac{1}{2}}, 0_{\frac{1}{2}}\}$. O conjunto de cociclos é $\{(t, u)_1 + (t, v)_{\frac{1}{4}} + (w, t)_{\frac{1}{2}}, (u, v)_1 + (t, v)_{\frac{1}{4}} + (v, w)_{\frac{1}{2}}, (t, u)_1 + (u, v)_1, (v, w)_{\frac{1}{2}} + (w, t)_{\frac{1}{2}}, (w, t)_{\frac{1}{2}} + (t, v)_{\frac{1}{4}} + (u, v)_1, (t, u)_1 + (t, v)_{\frac{1}{4}} + (v, w)_{\frac{1}{2}}, (w, t)_{\frac{1}{2}} + (t, u)_1 + (u, v)_1 + (v, w)_{\frac{1}{2}}, 0_{\frac{1}{4}}, 0_{\frac{1}{2}}\}$. O conjunto cociclo fuzzy é $\{(t, u)_1 + (w, t)_{\frac{1}{2}}, (u, v)_1 + (v, w)_{\frac{1}{2}}, (t, u)_1 + (u, v)_1, (v, w)_{\frac{1}{2}} + (w, t)_{\frac{1}{2}}, (w, t)_{\frac{1}{2}} + (u, v)_1, (t, u)_1 + (v, w)_{\frac{1}{2}}, (w, t)_{\frac{1}{2}} + (t, u)_1 + (u, v)_1 + (v, w)_{\frac{1}{2}}, 0_{\frac{1}{2}}\}$. O conjunto deciclos e o conjunto de cociclos não são e não geram o espaço de vetores sobre $(\mathcal{Z})_\infty$ por causa da presença de $0_{\frac{1}{2}}$ e $0_{\frac{1}{4}}$. Note também que no conjunto de ciclos, $((t, u)_1 + (u, v)_1 + (t, v)_{\frac{1}{4}}) + ((v, w)_{\frac{1}{2}} + (w, t)_{\frac{1}{2}} + (t, v)_{\frac{1}{4}}) = (t, u)_1 + (u, v)_1 + (v, w)_{\frac{1}{2}} + (w, t)_{\frac{1}{2}} + 0_{\frac{1}{4}} \neq (t, u)_1 + (u, v)_1 + (v, w)_{\frac{1}{2}} + (w, t)_{\frac{1}{2}}$. O conjunto dos ciclos fuzzy é um espaço vetorial sobre \mathcal{Z}_∞ neste exemplo. O conjunto dos cociclos fuzzy não é um espaço vetorial sobre \mathcal{Z}_∞ uma vez que $((v, w)_{\frac{1}{2}} + (w, t)_{\frac{1}{2}}) + ((w, t)_{\frac{1}{2}} + (t, u)_1 + (u, v)_1 + (v, w)_{\frac{1}{2}}) = (t, u)_1 + (u, v)_1 + 0_{\frac{1}{2}} \neq (t, u)_1 + (u, v)_1$.

Exemplo 3.7.4. (Moderson e Nair, 1996a) Seja V, σ, X como no Exemplo 3.7.3. Seja $X' = X \cup \{uw\}$. Definimos o conjunto fuzzy μ' de X' por $\mu' = \mu$ sobre X e $\mu'(u, w) = \frac{1}{8}$. Então o conjunto dos ciclos fuzzy e o conjunto dos cociclos fuzzy de (σ, μ') coincide com o conjunto dos ciclos e o conjunto dos cocilos de (σ, μ) do Exemplo 3.7.3, respectivamente.

Seja $CS(\sigma, \mu)$, $FCS(\sigma, \mu)$, $CoS(\sigma, \mu)$ e $FCoS(\sigma, \mu)$ denotando os conjuntos dos ciclos, dos ciclos fuzzy, dos cociclos e dos cociclos fuzzy de (σ, μ) , respectivamente.

Mostraremos agora que apesar de CS, FCS, CoS e FCoS não serem necessariamente espaços vetoriais sobre \mathcal{Z}_2 , eles estão bem próximo. De fato, é claro que os seguintes resultados que os conceitos de *twigs* (fuzzy) e *chords* (fuzzy) introduzidos aqui terão consequências similares aos de suas contrapartidas no caso crisp.

Claramente, CS, FCS, CoS, FCoS são subconjuntos de S_μ . Seja S um subconjunto de S_μ . Seja $\langle S \rangle$ denotando a interseção de todos os subsemigrupos de S_μ que contêm S . Então $\langle S \rangle$ é o menor subsemigrupo de S_μ que contêm S . Seja $S^+ = \{(x_1)_{t_1} + \dots + (x_n)_{t_n} | (x_i)_{t_i} \in S, i = 1, \dots, n, n \in \mathcal{N}\}$, onde \mathcal{N} denota o conjunto dos inteiros positivos. Então S^+ é um subsemigrupo de S_μ .

Teorema 3.7.1. (Moderson e Nair, 1996a) $\langle CS \rangle = (CS)^+ = CS \cup \{x_a + 0_b | x_a \in CS, 0_b \in (CS)^+\}$. $\langle CS \rangle$ têm 0_m como sua identidade, onde $m = \max\{b | 0_b \in (CS)^+\}$.

Prova: Uma vez que $(CS)^+$ é um subsemigrupo de S_μ que contem CS, $\langle CS \rangle \subseteq (CS)^+$. Contudo, está claro que $\langle CS \rangle \supseteq (CS)^+$ já que $\langle CS \rangle$ é próximo a $+$. Portanto $\langle CS \rangle = (CS)^+$. Obviamente $\langle CS \rangle \supseteq CS \cup \{x_a + 0_b | x_a \in CS, 0_b \in (CS)^+\}$. Agora $foot(CS)$ é um espaço vetorial sobre \mathcal{Z}_2 (Harary, 1972). Seja $z_t, y_s \in CS$. Então $z + y \in foot(CS)$. Também $z_t + y_s = (z + y)_r$, onde $r = \min\{t, s\}$. agora $z_t = (u_1, v_1)_{t_1} + \dots + (u_n, v_n)_{t_n}$ e $y_s = (p_1, q_1)_{s_1} + \dots + (p_m, q_m)_{s_m}$, onde $(u_i, v_i), (p_j, q_j) \in X, i = 1, \dots, n$ e $j = 1, \dots, m$. Seja $I = \{(u_i, v_i) | i = 1, \dots, n\} \cap \{(p_j, q_j) | j = 1, \dots, m\}$. Suponha que $I \neq \emptyset$. Rearranjando os termos na representação de z_t e y_s se necessário, temos $z_t + y_s = (u_1 v_1)_{t_1} + \dots + (u_{i-1} v_{i-1})_{t_{i-1}} + (p_1, q_1)_{s_1} + \dots + (p_{j-1}, q_{j-1})_{s_{j-1}} + 0_b$, onde $0_b = (u_i, v_i)_{t_i} + \dots + (u_n, v_n)_{t_n} + (p_j, q_j)_{s_j} + \dots + (p_m, q_m)_{s_m}$, $b = \min\{t_i, \dots, t_n, s_j, \dots, s_m\}$ e $I = \{(u_i, v_i), \dots, (u_n, v_n)\} = \{(p_j, q_j), \dots, (p_m, q_m)\}$. Agora $(u_1, v_1) + \dots + (u_{i-1}, v_{i-1}) + (p_1, q_1) + \dots + (p_{j-1}, q_{j-1}) \in foot(CS)$ já que $foot(CS)$ é um espaço vetorial sobre \mathcal{Z}_2 . Também $(z + y)_a = (u_1, v_1)_{t_1} + \dots + (u_{i-1}, v_{i-1})_{t_{i-1}} + (p_1, q_1)_{s_1} + \dots + (p_{j-1}, q_{j-1})_{s_{j-1}}$, onde $a = \min\{t_1, \dots, t_{i-1}, s_1, \dots, s_{j-1}\}$. Agora $(z + y)_r + 0_b = (z + y)_a + 0_b$. Com isso $z_t + y_s \in CS \cup \{x_a + 0_b | x_a \in CS, 0_b \in (CS)^+\}$. Isto é, a soma de quaisquer dois elementos de CS está em $CS \cup \{x_a + 0_b | x_a \in CS, 0_b \in (CS)^+\}$. O caso onde $I = \emptyset$ é similar. \square

Corolário 3.7.1. (Moderson e Nair, 1996a) $\langle FCS \rangle = (FCS)^+ = FCS \cup \{x_a + 0_b | x - a \in FCS, 0_b \in (FCS)^+\}$. $\langle FCS \rangle$ têm 0_m como sua identidade, onde $m = \max\{b | 0_b \in (FCS)^+\}$.

Prova: FCS é o conjunto dos ciclos de (σ, μ_E) . \square

De modo similar, obtemos os dois seguintes resultados.

Teorema 3.7.2. (Moderson e Nair, 1996a) $\langle CoS \rangle = (CoS)^+ = CoS \cup \{x_a + 0_b \mid x_a \in CoS, 0_b \in (CoS)^+\}$. $\langle CoS \rangle$ têm 0_m como sua identidade, onde $m = \max\{b \mid 0_b \in (CoS)^+\}$.

Definição 3.7.9. (Moderson e Nair, 1996a) O posto do ciclo de (σ, μ) , escrito $m(\sigma, \mu)$, é definido como

$$m(\sigma, \mu) = \max\left\{\sum_{i=1}^n t_i \mid (x_i)_{t_i} \in CS, i = 1, \dots, n, \{x_1, \dots, x_n\} \text{ é a base para } foot(CS)\right\}.$$

O posto do ciclo fuzzy de (σ, μ) , escrito $fm(\sigma, \mu)$ é definido pelo posto do ciclo de (σ, μ_E) . Se $\{x_1, \dots, x_n\}$ é uma base para $foot(CS)$ tal que $m(\sigma, \mu) = \sum_{i=1}^n t_i$, onde $(x_i)_{t_i} \in CS$, $i = 1, \dots, n$ então $\{x_1, \dots, x_n\}$ é chamada de base ciclo de $\langle CS \rangle$. Se $\{x_1, \dots, x_n\}$ é uma base para $foot(FCS(\sigma, \mu))$ tal que $fm(\sigma, \mu) = \sum_{i=1}^n t_i$, onde $(x_{t_i}) \in FCS(\sigma, \mu)$, $i = 1, \dots, n$ então $\{x_1, \dots, x_n\}$ é chamada de base de ciclo fuzzy de $\langle CS(\sigma, \mu) \rangle$.

Teorema 3.7.3. (Moderson e Nair, 1996a) Seja $\{x_1, \dots, x_n\}$ uma base ciclo de $\langle CS \rangle$. Então $\forall x_t \in CS$, existe um reordenamento de x_1, \dots, x_n tal que $x_t = (x_1)_{t_1} + \dots + (x_m)_{t_m}$, $m \leq n$, onde $t_i = \mu(x_i)$, $i = 1, \dots, m$.

Prova: Como $\{x_1, \dots, x_n\}$ é uma base para $foot(CS)$, existe um reordenamento de x_1, \dots, x_n tal que $x = x_1 + \dots + x_m$, $m \leq n$. Suponha que $t > \min\{t_1, \dots, t_m\}$. Então existe um t_i , $i = 1, \dots, m$ tal que $t > t_i$. Agora $x \notin \langle \{x_1 + \dots + x_n\} \setminus \{x_i\} \rangle$. Portanto $(\{x_1 + \dots + x_n\} \setminus \{x_i\}) \cup \{x\}$ é uma base para $foot(CS)$. Contudo, isto contradiz a hipótese de que $\{x_1, \dots, x_n\}$ é uma base ciclo de $\langle CS \rangle$ já que $t > t_i$. Portanto $t \leq \min\{t_1, \dots, t_m\}$. Agora $x = u_1 v_1 + \dots + u_r v_r$, onde $u_i v_i \in X$, $i = 1, \dots, r$. Portanto $x_t = (u_1 v_1)_{a_1} + \dots + (u_r v_r)_{a_r}$ e $(x_1)_{t_1} + \dots + (x_m)_{t_m} = (u_1 v_1)_{a_1} + \dots + (u_r v_r)_{a_r} + 0_a$ para algum $a \in (0, 1]$, onde $\mu(u_i v_i) = a_i$, $i = 1, \dots, r$ e $\min\{t_1, \dots, t_m\} = \min\{a_1, \dots, a_r, a\}$. Agora $\min\{a_1, \dots, a_r, a\} \leq \min\{a_1, \dots, a_r\} = t \leq \min\{t_1, \dots, t_m\} = \min\{a_1, \dots, a_r, a\}$. Portanto $t = \min\{t_1, \dots, t_m\}$ e então $x_t = (x_1)_{t_1} + \dots + (x_m)_{t_m}$. \square

Corolário 3.7.2. (Moderson e Nair, 1996a) Seja $\{x_1, \dots, x_n\}$ uma base de ciclos fuzzy de $\langle CS(\sigma, \mu) \rangle$. Então $\forall x_t \in FCS(\sigma, \mu)$, existe um reordenamento de x_1, \dots, x_n tal que $x_t = (x_1)_{t_1} + \dots + (x_m)_{t_m}$, $m \leq n$, onde $t_i = \mu(x_i)$, $i = 1, \dots, m$.

Prova: FCS é o conjunto de ciclos de (σ, μ_E) e o posto do ciclo fuzzy de (σ, μ) é o posto do ciclo de (σ, μ_E) . \square

Definição 3.7.10. (Moderson e Nair, 1996a) O posto do cociclo de (σ, μ) escrito $m_c(\sigma, \mu)$ é definido como

$$m_c(\sigma, \mu) = \max \left\{ \sum_{i=1}^n t_i \mid (x_i)_{t_i} \in CoS, i = 1, \dots, n, \{x_1, \dots, x_n\} \text{ é uma base para } foot(CoS) \right\}.$$

O posto dos cociclos fuzzy de (σ, μ) , escrito $fm_c(\sigma, \mu)$ é definido como posto do ciclo (σ, μ_E) . Se $\{x_1, \dots, x_n\}$ é uma base para $foot(CoS)$ tal que $m_c(\sigma, \mu) = \sum_{i=1}^n t_i$, onde $(x_i)_{t_i} \in CoS$, $i = 1, \dots, n$ então $\{x_1, \dots, x_n\}$ é chamado uma base cociclo de $\langle CoS \rangle$. Se $\{x_1, \dots, x_n\}$ é uma base para $foot(FCoS(\sigma, \mu))$ tal que $fm_c(\sigma, \mu) = \sum_{i=1}^n t_i$ onde $(x_i)_{t_i} \in FCoS(\sigma, \mu)$, $i = 1, \dots, n$ então $\{x_1, \dots, x_n\}$ é chamada de base de cociclos fuzzy de $\langle CS(\sigma, \mu) \rangle$.

De modo similar, obtemos os dois seguintes resultados:

Teorema 3.7.4. (Moderson e Nair, 1996a) Seja $\{x_1, \dots, x_n\}$ uma base de cociclos de $\langle CoS \rangle$. Então $\forall x_t \in CoS$, existe um reordenamento de x_1, \dots, x_n tal que $x_t = (x_1)_{t_1} + \dots + (x_m)_{t_m}$, $m \leq n$ onde $t_i = \mu(x_i)$, $i = 1, \dots, m$.

Corolário 3.7.3. (Moderson e Nair, 1996a) Seja $\{x_1, \dots, x_n\}$ uma base de cociclos fuzzy de $\langle CS(\sigma, \mu) \rangle$. Então $\forall x_t \in FCS(\sigma, \mu)$ existe um reordenamento de x_1, \dots, x_n tal que $x_t = (x_1)_{t_1} + \dots + (x_m)_{t_m}$, $m \leq n$ onde $t_i = \mu(x_i)$, $i = 1, \dots, m$.

3.8 Métrica em Grafos Fuzzy

Uma definição elegante de uma métrica em um grafo fuzzy foi dado em (Rosenfeld, 1975). Se ρ é um caminho consistindo de nós x_0, x_1, \dots, x_n em um grafo fuzzy $G = (\sigma, \mu)$, o μ -comprimento de ρ é definido por

$$l(\rho) = \sum_{i=1}^n \mu(x_{i-1}, x_i)^{-1} \quad (3.2)$$

Se $n = 0$ então $l(\rho)$ é escolhido ser igual a zero. Agora para dois nós $x, y \in G$, a μ -distância $\delta(x, y)$ é definida como sendo o mínimo dos μ -comprimentos de todos os caminhos que unem x a y .

Proposição 3.8.1. (Rosenfeld, 1975) $\delta(x, y)$ é uma métrica.

Prova:

- (a) $\delta(x, y) = 0$ se e somente se $x = y$, já que $l(p) = 0$ se, e somente se, ρ tem comprimento 0.
- (b) $\delta(x, y) = \delta(y, x)$ já que o reverso de um caminho é um caminho e μ é simétrico.
- (c) $\delta(x, z) \leq \delta(x, y) + \delta(y, z)$, já que a concatenação de um caminho de x para y com um caminho de y para z é um caminho de x para z e l é aditivo para concatenação de caminhos.

□

A seguir, vamos introduzir algumas definições baseadas nesta métrica. Suponha que $G = (\sigma, \mu)$ é um grafo fuzzy com S sendo o conjunto de nós. A *excentricidade* $e(v)$ de um nó $v \in S$ é definida como sendo o máximo de todas as μ -distâncias $\delta(v, w)$ para todos $w \in S$. Um *centro* de um grafo fuzzy conectado é um nó cuja excentricidade é mínima. O *raio* de um grafo fuzzy conectado é o mínimo de todas as excentricidades dos nós do grafo fuzzy. Estas definições são analogias fuzzy das definições correspondentes na teoria de grafos (crisp) com importantes aplicações em telecomunicações. Usando o fato de que δ é uma métrica, o resultado seguinte é obtido facilmente e portanto omitimos os detalhes.

Proposição 3.8.2. (Bhattacharya, 1987) Para um grafo fuzzy conectado, temos

$$\text{raio}(G) \leq \text{diametro}(G) \leq 2 \times \text{raio}(G). \quad (3.3)$$

Exemplo 3.8.1. (Bhattacharya, 1987) Considere o grafo definido no Exemplo 3.4.1. Temos $\delta(x, y) = 1$, $\delta(x, z) = 2$, $\delta(y, z) = 2$. Portanto $e(x) = e(y) = e(z) = 2$ e todos os nós do grafo são centros. Observamos que este grafo não possui nós de corte.

Exemplo 3.8.2. (Bhattacharya, 1987) Considere o grafo definido no Exemplo 3.4.2. Observamos que $e(x) = 5 = e(w) = e(u)$ e $e(z) = 4$. Portanto z é o centro deste grafo. Observamos também que este grafo não possui nós de corte.

Dos Exemplos 3.8.1 e 3.8.2 e também procurando obter um resultado padrão da teoria de grafos crisp, pode ser tentado conjecturar que para um grafo fuzzy G , um centro não pode ser um nó de corte de G . Contudo, o seguinte exemplo mostra que isto não é verdade em geral.

Exemplo 3.8.3. (Bhattacharya, 1987) Considere o grafo fuzzy dado na Figura 3.14. Aqui $\mu(x, u) = 0.9$, $\mu^\infty(x, u) = 0.25$. Então w é um nó de corte. Contudo w é também um centro uma vez que $e(w) = 4$, $e(u) = 4$, $e(y) = 7$, $e(x) = 6$, $e(z) = 7$, $e(v) = 5$.

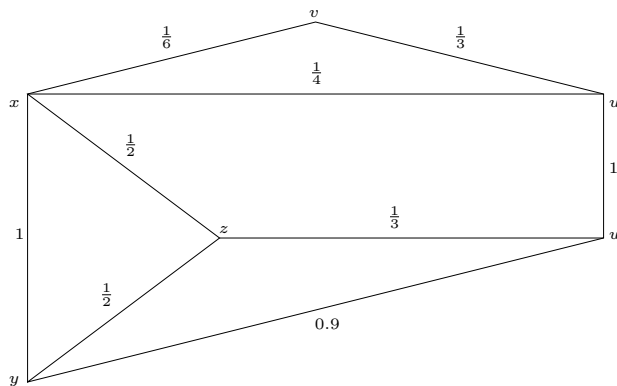


Figura 3.14: Exemplo de centro em um grafo fuzzy

3.9 Grupos fuzzy de um grafo fuzzy

O objetivo desta seção é obter uma analogia fuzzy de um resultado básico na teoria de grafos onde dado um grafo, pode-se associar um grupo em um caminho natural.

Definição 3.9.1. (Bhattacharya, 1987) Seja $G = (\sigma, \mu)$ um grafo fuzzy com S como conjunto subjacente. Um mapeamento $\phi : S \rightarrow S$ é um morfismo de G se

$$\mu(\phi(x), \phi(y)) \geq \mu(x, y) \quad \forall x, y \in S \quad (3.4)$$

$$\sigma(\phi(x)) \geq \sigma(x) \quad \forall x \in S \quad (3.5)$$

Proposição 3.9.1. (Bhattacharya, 1987) Dado um grafo fuzzy $G = (\sigma, \mu)$ seja G^* o conjunto de todos os morfismos de G . Se ϕ, ψ são dois elementos em G^* , seja $\phi \circ \psi$ denotando o produto usual do mapeamento ϕ, ψ . Então (G^*, \circ) é um semigrupo.

Prova: Se ϕ, ψ são dois elementos de G^* então para $x, y \in S$ temos

$$\begin{aligned} \mu((\phi \circ \psi)(x), (\phi \circ \psi)(y)) &= \mu(\phi(\psi(x)), \phi(\psi(y))) \\ &\geq \mu(\psi(x), \psi(y)), \text{ usando 3.4} \\ &\geq \mu(x, y), \text{ usando 3.4} \end{aligned}$$

Novamente, para $x \in S$ temos

$$\sigma((\phi \circ \psi)(x)) \geq \sigma(\psi(x)) \geq \sigma(x),$$

usando 3.5. Por isso $\phi \circ \psi \in G^*$. O fato é associado diretamente da propriedade padrão do conjunto produto (*set-theoretic product*) de dois mapeamentos. Portanto $(G, *)$ é um semigrupo. \square

Corolário 3.9.1. (*Bhattacharya, 1987*) Com a mesma hipótese da Proposição 3.9.1, seja $e : S \rightarrow S$ um mapeamento definido por $e(s) = s$ para todo $x \in S$. Então $e \in G^*$ e para todo $\phi \in G^*$ temos

$$(i) \quad \phi \circ e = \phi = e \circ \phi$$

$$(ii) \quad \lambda(\phi) \geq \lambda(e)$$

Portanto e serve como o elemento identidade do semigrupo (G^*, \circ) .

Mostramos agora como associar um grupóide fuzzy com qualquer grafo fuzzy em um caminho natural.

Proposição 3.9.2. (*Bhattacharya, 1987*) Seja $G = (\sigma, \mu)$ um grafo fuzzy e G^* o conjunto de todos os morfismos de G . Definimos um mapeamento $\lambda : \rightarrow [0, 1]$ por

$$\lambda(\phi) = \sup\{\mu(\phi(x), \phi(y)) : (x, y) \in S \times S\} \quad (3.6)$$

para todo $\phi \in G^*$. Então λ é um subgrupóide sobre G^* .

Prova: Pela Proposição 3.9.1, (G^*, \circ) é um semigrupóide. Seja $\phi, \psi \in G^*$. Então

$$\begin{aligned} \lambda(\phi \circ \psi) &= \sup\{\mu(\phi \circ \psi(x), \phi \circ \psi(y)) : (x, y) \in S \times S\} \\ &\geq \sup\{\mu(\psi(x), \psi(y)) : (x, y) \in S \times S\}, \text{ usando 3.4} \\ &= \lambda(\psi), \text{ usando 3.4} \end{aligned}$$

Portanto, temos

$$\lambda(\phi \circ \psi) \geq \lambda(\psi)$$

Consequentemente, $\lambda(\phi \circ \psi) \geq \lambda(\phi) \wedge \lambda(\psi)$. Portanto λ é um subgrupóide sobre G^* . \square

O conceito de morfismo de um grafo fuzzy é estendido a seguir.

Definição 3.9.2. (Bhattacharya, 1987) *Seja $G = (\sigma, \mu)$ um grafo fuzzy com S como o conjunto subjacente (o conjunto de nós de G). Uma designação (one-to-one), um mapeamento $\phi : S \rightarrow S$ é um automorfismo de G se*

$$\mu(\phi(x), \phi(y)) = \mu(x, y) \quad \forall x, y \in S \quad (3.7)$$

$$\sigma(\phi(x)) = \sigma(x) \quad \forall x \in S \quad (3.8)$$

A prova do resultado seguinte é similar a prova da Proposição 3.9.1 e omitiremos os detalhes.

Proposição 3.9.3. (Bhattacharya, 1987) *Seja G_* o conjunto de todos os automorfismos de um grafo fuzzy $G = (\sigma, \mu)$. Então G_* é um grupo sob o conjunto produto de mapeamentos como a operação binária.*

No resultados seguinte mostramos como associar um grupo fuzzy com grafo fuzzy.

Teorema 3.9.1. (Bhattacharya, 1987) *Seja $G = (\sigma, \mu)$ um grafo fuzzy e G_* o grupo de todos os automorfismos de G . Definimos um mapeamento $\tau : G_* \rightarrow [0, 1]$ por*

$$\tau(\phi) = \sup\{\mu(\phi(x), \phi(y)) : (x, y) \in S \times S\} \quad (3.9)$$

para todo $\phi \in G_*$. Então τ é um grupo fuzzy sobre G_* .

Prova: Notamos que a definição de τ dado por 3.9 é idêntica a definição de λ dada por 3.6 na construção do subgrupóide fuzzy. Uma vez que $\phi \in G_*$, usando 3.7, a equação 3.9 pode ser expressa como:

$$\tau(\phi) = \sup\{\mu(x, y) : (x, y) \in S \times S\} \quad (3.10)$$

De (3.10) se $\phi, \psi \in G_*$, então

$$\tau(\phi \circ \psi) \geq \tau(\phi) \wedge \tau(\psi)$$

e

$$\tau(\phi^{-1}) = \tau(\phi).$$

Portanto τ é um grupo fuzzy sobre G_* . □

Observação: A motivação por trás do Teorema 3.9.1 é induzir o grupóide fuzzy λ definido sobre G_* a um grupo fuzzy impondo algumas condições extras sobre a definição de morfismo de um grafo fuzzy e obter uma definição plausível do automorfismo de um grafo fuzzy. Contudo, se for tentado uma definição mais fraca de automorfismo como dada aqui, então o conjunto dos automorfismos definidos formam um grupo. Consequentemente não é possível construir um grupo fuzzy fora disso.

Capítulo 4

Problemas de grafos fuzzy e suas soluções

Quando citamos um problema de grafos fuzzy, a característica fuzzy do problema pode ser encontrada em diversos níveis: da estrutura do grafo (nós e arcos) aos parâmetros associados ao grafo. Ainda quando considera-se fluxo em uma rede, pode-se associar uma incerteza a este valor. Também com relação à solução destes problemas existem diferentes formatos apresentados na literatura, dependendo do objetivo do decisor.

Durante a pesquisa bibliográfica realizada foram encontrados dois tipos principais de problemas de grafos fuzzy, que estaremos apresentando neste capítulo. Para outras abordagens, (Blue et al., 1997) propõe uma taxonomia, mas os tipos de problemas relatados podem ser reduzidos aos dois casos que iremos descrever. Quanto às soluções, estas serão apresentadas para cada tipo de problema, dados nas seções que seguem.

4.1 Problemas com estrutura do grafo crisp e parâmetros fuzzy

Este é o problema mais citado na literatura. São problemas em que a estrutura do grafo é bem conhecida e os parâmetros associados são representados por números fuzzy. São vários os exemplos, principalmente considerando custo ou tempo como números fuzzy (vide Figura 4.1).

O caminho mínimo com custo fuzzy é o problema de grafos com parâmetros fuzzy mais estudado. Para conhecer melhor a bibliografia sobre este problema, veja as Seções 5.2 e 5.3, respectivamente.

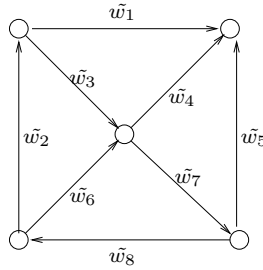


Figura 4.1: Exemplo de grafo com parâmetros fuzzy

Problemas em que a capacidade dos arcos é um número fuzzy são os problemas de fluxos em redes. No problema de fluxo máximo é associado um grau de confiabilidade da solução dado o fluxo nos arcos (vide Capítulo 8) para fluxos com valores inteiros ou contínuos. Este problema foi bastante estudado por Chanas (1987) que pode ser visto na Seção 8.1. Um problema mais complexo envolve parâmetros fuzzy tanto na capacidade dos arcos quanto nos custos associados, que é o caso do problema de fluxo de custo mínimo. Este estudo pode ser encontrado no Capítulo 9.

Os problemas da árvore geradora mínima e de emparelhamento podem ter seus parâmetros fuzzy, embora até o momento poucos estudos foram encontrados na literatura. Neste trabalho foram realizados estudos nos Capítulos 6 e 7.

Quanto à solução, de modo geral, aquela proposta por (Bellman e Zadeh, 1970) é a mais utilizada para o problema de grafos com parâmetros fuzzy e será definida na próxima seção. Aqui também será apresentada abordagem adotada por Okada (2001) para o problema do caminho mínimo com parâmetros fuzzy e que será a referência à nossa proposta de solução para os problemas de mesma natureza.

4.1.1 Abordagem de Bellman e Zadeh, 1970

O trabalho de Bellman e Zadeh (1970), traz as seguintes definições:

Definição 4.1.1. (Bellman e Zadeh, 1970) *Assumindo que são dados um objetivo fuzzy (C) e restrições fuzzy (R) em um espaço de alternativas X , então, C e R combinam para formar uma decisão, D , que é um conjunto fuzzy resultante da intersecção de C e R . Em símbolos*

$$D = C \cap R$$

e $\mu_D = \mu_C \wedge \mu_R$.

A Figura 4.2 é um exemplo desta definição.

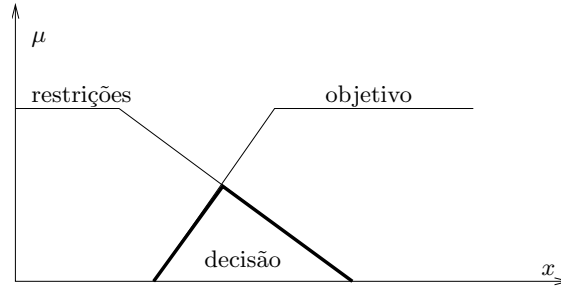


Figura 4.2: Exemplo de um gráfico do modelo de Bellman e Zadeh

Portanto, considerando uma decisão *ótima*, temos:

$$\mu_{D^{\max}} = \max_{x_v} \{ \mu_C(x_v) \wedge \mu_R(x_v) \}$$

onde x_v é uma solução factível para o problema em grafos. Na Figura 4.2, dada a função de pertinência μ_D formada pela intersecção das funções de pertinência das restrições (μ_R) e do objetivo (μ_C), o valor calculado para ser a decisão ótima é dado pelo valor modal. Na literatura é comum encontrar trabalhos que tratam os problemas sob esta abordagem.

4.1.2 Abordagem de Okada, 2001

Seja, por exemplo, um problema de árvore geradora mínima com parâmetros fuzzy (Capítulo 6), isto é, em um grafo G com custos fuzzy \tilde{c}_{ij} . Temos que encontrar uma árvore geradora tal que o custo seja *mínimo*. Seja o exemplo da Figura 4.3:

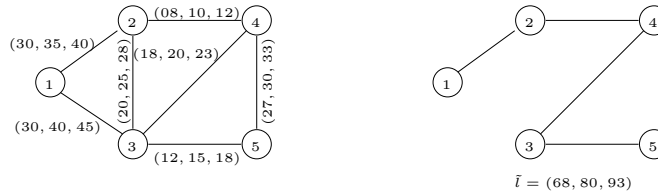


Figura 4.3: Exemplo de uma rede fuzzy

onde os números $(\underline{a}, a, \bar{a})$ são números triangulares fuzzy. Temos, por um trabalho encontrado para este problema (Chang e Lee, 1999), que a solução é dada pela árvore apresentada na Figura 4.3 para o caso pessimista, otimista e neutro. Basicamente, cada caso é baseado em nos valores \underline{a} , a e \bar{a} que representam o número fuzzy. Este trabalho calcula um valor crisp associado ao

número fuzzy para cada caso, e com isso obtém três problemas críps que são resolvidos através de algoritmos clássicos. Apesar de ter sido encontrado apenas um trabalho para o caso do problema da árvore geradora mínima, diversos trabalhos são apresentados com a proposta de encontrar a melhor solução para outros problemas com parâmetros fuzzy baseando-se em uma ordenação dos valores. Vamos tomar as árvores geradoras mostradas na Figura 4.4:

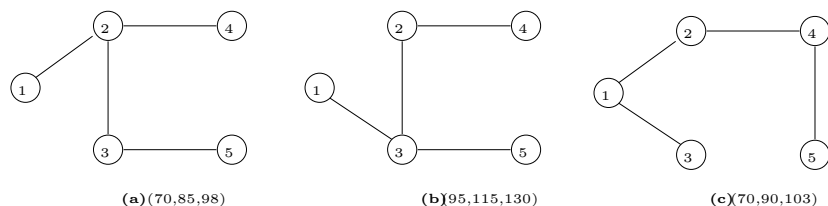


Figura 4.4: Exemplo de árvores geradoras para a Figura 4.3

Estas foram construídas a partir da rede G dada na Figura 4.3. Agora, observe os comprimentos fuzzy associados em comparação à solução obtida na mesma Figura 4.3:

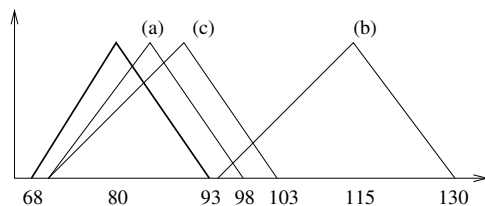


Figura 4.5: Comprimento das árvores dadas na Figura 4.4

É fácil ver que o comprimento fuzzy associado a (b) é maior do que o custo da solução encontrada previamente por (Chang e Lee, 1999), mas as outras árvores têm um considerável *valor de possibilidade* de ser menor do que a solução encontrada. Para o caso do problema da árvore geradora mínima, a solução não é determinada pelo nível de satisfação, ou pelo nível de violação de determinada restrição, mas por uma comparação entre as possíveis soluções. Neste caso, devolver apenas a solução de menor custo com maior grau de satisfação, dada alguma ordenação, poderia ser vista como perda de informação. Um conjunto solução onde cada árvore geradora (ou caminho entre dois nós) possui um grau de pertinência associado ao conjunto pode ser muito proveitoso ao decisor.

Com base neste fato, os algoritmos para os problemas da árvore geradora mínima fuzzy e do caminho mínimo fuzzy foram construídos de modo a fornecer todo o conjunto solução para o problema ($\mathcal{T} = \{T^i, i = 1, \dots, n\}$, ou seja uma árvore T^i com grau de pertinência μ_i ao con-

junto solução). Também foi construído o algoritmo para o caso do fluxo máximo com estrutura e solução crisp, mas com capacidade fuzzy. A construção destes algoritmos é intuitivamente simples, mas com dificuldades intrínsecas que veremos a seguir.

Chanas (1987), Okada e Soper (2000), Okada (2001) utilizam a teoria de possibilidade (Zadeh, 1978) para calcular a pertinência de determinada solução para o caminho mínimo. (Blue et al., 1997) também faz o uso desta teoria, embora não explicita isso em seu trabalho. Para construir a solução do problema teríamos que:

1. encontrar todas as soluções candidatas (árvores, caminhos, emparelhamento, etc.) e;
2. calcular o valor de possibilidade entre as soluções encontradas.

O item 1 já é um problema combinatorial. Após resolver este problema ainda teremos que compará-los para calcular o grau de possibilidade de cada candidato a participar da solução: um número que pode ser computacionalmente inviável de calcular.

Portanto, os algoritmos devem ser construídos de modo a evitar cálculos desnecessários, por exemplo tentando construir apenas as soluções que possuem valor de possibilidade não-nula adotando testes de corte da solução com base em propriedades de cada problema e/ou *ordenando* a forma de comparações para que uma solução que possua valor de possibilidade nula seja descartada previamente. Além disso, podemos considerar formas de armazenamento das informações, dentre outros detalhes que facilitariam a resolução do problema.

Mesmo com estes procedimentos, os algoritmos mostraram serem fortemente influenciados pela incerteza dos dados, como já havia sido constatado em Okada e Soper (2000) para o problema do caminho mínimo com parâmetros fuzzy. Mesmo com os cortes dos elementos que possuem valor de possibilidade nulo (por conseqüência, pertinência nula), se a incerteza dos dados for significativa, o número de soluções será alto e será um fator complicante no cálculo do conjunto de soluções. Um exemplo utilizado por Okada e Soper (2000) será visto no Capítulo 5. Ao buscar o menor caminho entre Boston e Los Angeles tendo números trapezoidais com platô de 20% e desvio de 10%, teríamos 653 caminhos mínimos com grau de pertinência igual a 1. Mesmo que existam $2,52 \times 10^{15}$ elementos candidatos para este problema e que apenas $1,14 \times 10^{-4}\%$ façam parte da solução, ainda assim a incerteza acentua a parte combinatorial e torna os algoritmos não eficientes para estes casos.

Para viabilizar a obtenção de uma solução, mesmo que aproximada, foram estudadas algumas heurísticas e metaheurísticas que utilizassem as características do problema. Considerando que todas as soluções podem participar da solução geral do problema, se o valor de possibilidade for não nulo, temos que:

- não estamos interessados em um único elemento, mas em um conjunto;
- pensando em termos dos valores de possibilidade para os arcos (Okada, 2001), os elementos que possuem os maiores valores de possibilidade são os mais significativos;
- a diversidade dos elementos analisados é importante para garantir uma solução mais próxima da *ótima*.

Com base nestas características, foi implementado um algoritmo genético (Michalewicz, 1996) utilizando trabalhos encontrados na literatura para problemas com restrições. Mas ainda encontramos problemas:

- Criar e manter a diversidade da população.
- Codificação de alguns problemas, como o caminho mínimo : tamanho do cromossomo não seria fixo.

Os resultados para estes problemas têm sido animadores e serão analisados nos respectivos capítulos. Um apêndice com os conceitos básicos de algoritmos genéticos se encontra no final deste trabalho (Apêndice B).

4.2 Problemas com estrutura de grafos fuzzy

Neste caso o grafo $\tilde{G} = (\tilde{N}, \tilde{A})$ não é bem conhecido. Valores de pertinência são associados aos arcos $(i, j) \in \tilde{A}$ e/ou aos nós $i \in \tilde{N}$, como descrito na definição de Rosenfeld (1975). No Capítulo 3, as propriedades de um grafo fuzzy são estudadas.

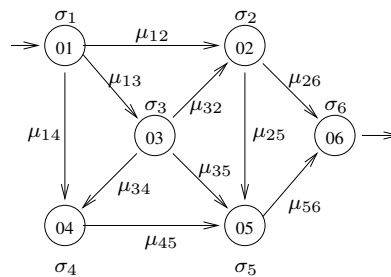


Figura 4.6: Exemplo de um grafo com estrutura fuzzy

Para este tipo de problema, o trabalho de Delgado et al. (1990) é um dos mais importantes pois estabelece um algoritmo fundamental para os problemas de grafos em que os conjuntos de nós e arcos são conjuntos fuzzy.

Outros trabalhos foram baseados em (Delgado et al., 1990), dentre eles o de Chanas et al. (1995) para o problema do fluxo ótimo em estrutura de grafos fuzzy, que foi utilizado para resolver um problema de transporte. Também foi encontrado um trabalho para o problema da árvore geradora mínima por Chunde (1996). Na Seção a seguir, o algoritmo fundamental de Delgado et al. (1990) é apresentado.

4.2.1 O trabalho de Delgado et. al, 1990

Seja $\tilde{G} = (\sigma, \mu)$ um grafo fuzzy. σ é a função de pertinência associada ao conjunto de nós N e μ é a função de pertinência associada ao conjunto de arcos A , sendo que o valor de pertinência de um arcos (i, j) não pode exceder o valor de pertinência dos nós i e j , ou seja $\mu_{ij} \leq \sigma_i \wedge \sigma_j$ (Rosenfeld, 1975).

Seja um α -corte de \tilde{G} definido por $G^\alpha = (N^\alpha, A^\alpha)$. O grafo resultante é um grafo crisp onde

$$N^\alpha = \{i \in N | \sigma_i \geq \alpha\}$$

$$A^\alpha = \{(i, j) \in A | \mu_{ij} \geq \alpha\}$$

Como os conjuntos de nós e arcos são finitos, então um grafo fuzzy possui um número finito de α -cortes. De fato, existem uma seqüência de valores $\mathcal{A} = \{\alpha_1, \dots, \alpha_k\}$ para os quais os α -cortes não mudam nestes intervalos:

$$\begin{aligned} G^1 : & \quad I_1 = (0, \alpha_1] \\ G^2 : & \quad I_2 = (\alpha_1, \alpha_2] \\ & \quad \vdots \\ G^k : & \quad I_k = (\alpha_{k-1}, \alpha_k] \end{aligned}$$

Portanto, se para cada α -corte for resolvido um problema (crisp), a solução encontrada será ótima para todo o intervalo I_α .

Esta é a idéia principal do artigo e converge para um algoritmo que pode ser aplicado para qualquer problema em um grafo fuzzy:

O algoritmo fundamental de Delgado et. al, 1990

Para $p = 1$ até k , faça:

Calcule a solução utilizando G^{α_p} para

$$\forall (i, j) \in A^{\alpha_p}; c_{ij}(\beta); \forall \beta \in I_p$$

como função custo.

No artigo, considera-se a função custo c_{ij} associada ao grafo dependente do da função de pertinência. Em outros problemas, a função custo pode ser constante.

A solução fuzzy dada por este algoritmo pode ser apresentada de duas formas:

1. A solução fuzzy pode ser um conjunto

$$\mathcal{S} = \left\{ \frac{S_i}{\alpha_i}, i = 1, \dots, k \right\}$$

onde S_i é a solução do problema com valor de pertinência α_i ao conjunto solução.

2. A solução fuzzy pode ser um subgrafo $F = (\tau, \nu)$, sendo que

$$\nu_{ij} = \max_{(i,j) \in S_k} \{ \alpha_k \}$$

e

$$\tau_i = \max_{j \in N} \{ \nu_{ij} \vee \nu_{ji} \}$$

ou seja, o valor de pertinência de um arco no subgrafo F é igual ao da solução com o maior valor de pertinência que contém este arco e o valor de pertinência do nó é igual ao maior valor de pertinência dos arcos incidentes.

O primeiro tipo de solução fuzzy será utilizado nesta dissertação.

O algoritmo fundamental de Delgado et al. (1990) será adaptado e analisado para cada problema nos capítulos a seguir. Este algoritmo também será analisado em relação ao tamanho do conjunto solução e, para este caso, heurísticas serão propostas em cada capítulo.

PARTE II: ALGORITMOS

Capítulo 5

Problema de Caminhos Mínimos Fuzzy

Este é um dos problemas mais antigos e mais *importantes* 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. Em (Gondran e Minoux, 1986), (Goldbarg e Luna, 2000), (Wilson e Watkins, 1990), e (Ahuja et al., 1993), podem ser obtidas mais informações sobre o problema clássico e seus algoritmos.

5.1 Definição do problema do caminho mínimo clássico

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

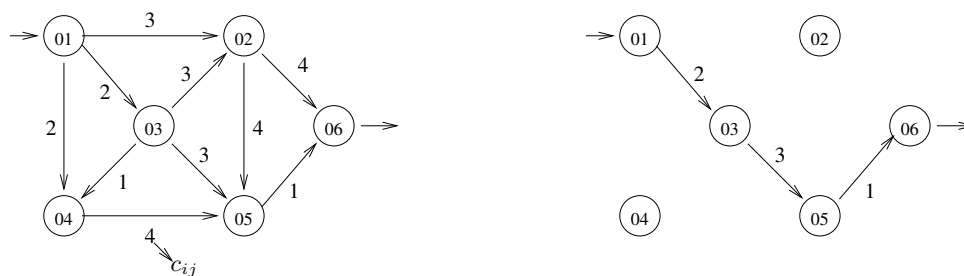


Figura 5.1: Exemplo de um problema de caminho mínimo

parâmetro pode ser interpretado como comprimento, custo ou tempo para percorrer um determinado arco (i, j) . O *comprimento* de um caminho P do grafo G é dado por: $l_P = \sum_{(i,j) \in P} c_{ij}$.

O problema do caminho mínimo pode consistir em:

- encontrar o caminho de menor comprimento (ou custo) no grafo entre o nó 1 e o nó m (os nós podem ser ordenados de forma conveniente). No exemplo dado na Figura 5.1 foi encontrado o caminho mínimo entre o nó 1 e o nó 6.
- Encontrar um caminho mínimo entre um nó e todos os outros nós.
- Encontrar um caminho mínimo entre todos os nós.

Neste trabalho devemos estudar os dois primeiros problemas.

Como em (Ahuja et al., 1993), 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, isto é, $c_{ij} > 0$.
3. O grafo é direcionado.

5.1.1 Formulação do problema do caminho mínimo

Considerando o grafo G descrito na Seção 5.1, seja

$$x_{ij} = \begin{cases} 1, & \text{se } (i, j) \in P \\ 0, & (i, j) \notin P \end{cases}$$

O problema do caminho mínimo (de um nó origem a um nó destino) pode ser formulado como um problema de fluxos em redes:

$$\text{Min} \quad \sum_{(i,j) \in A} c_{ij} x_{ij} \quad (5.1)$$

s.a.

$$\sum_{j:(i,j) \in A} x_{ij} - \sum_{k:(k,i) \in A} x_{ki} = \begin{cases} 1 & \text{se } i = 1 \\ 0 & \text{se } 2 \leq i \leq m - 1 \\ -1 & \text{se } i = m \end{cases} \quad (5.2)$$

$$x_{ij} \geq 0, \quad \forall (i, j) \in A \quad (5.3)$$

sendo que a restrição (5.3) foi relaxada devido às propriedades de unimodularidade e conservação de fluxo do problema (Bazaraa et al., 1990).

Os algoritmos não são derivados do método Simplex (Bazaraa et al., 1990). Muitas vezes, o problema dual associado:

$$\text{Max} \quad w_1 - w_m \quad (5.4)$$

$$\text{s.a.} \quad w_i - w_j \leq c_{ij} \quad i, j = 1, 2, \dots, m \quad (5.5)$$

$$w_i \text{ irrestrito} \quad i = 1, 2, \dots, m \quad (5.6)$$

é utilizado na resolução, como no algoritmo de Dijkstra (Goldbarg e Luna, 2000).

5.2 O problema de caminho mínimo fuzzy e os estudos encontrados na literatura

Para problemas com a estrutura e parâmetros bem definidos (crisp), existem algoritmos que encontram o caminho de menor custo (ou comprimento) de forma eficiente (Ahuja et al., 1993). Contudo, é comum encontrarmos problemas onde os parâmetros e/ou a própria estrutura possuem incertezas. Por exemplo, em um roteiro de viagem, o tempo de traslado de uma cidade (nó) i a uma cidade j pode ser considerado um parâmetro sujeito a incertezas: congestionamento, condições da rodovia, condições climáticas, disposição do motorista, etc. Da mesma forma, se imaginarmos a segurança de uma conexão em uma rede, uma determinada conexão pode ter um custo *mais atrativo* que outra, mas a primeira pode ser *mais instável* que a segunda.

Portanto, um problema pode ter diferente níveis de incerteza. Existem dois problemas que são citados pela maioria dos 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 pode ser 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.

No exemplo da Figura 5.2, temos um grafo em que os arcos possuem comprimento fuzzy. Por exemplo, no arco (1,2) temos que o comprimento é igual a $\tilde{c}_{12} = \{\frac{2}{0,8}, \frac{3}{0,3}\}$, ou seja o arco (1,2) tem comprimento 2 com valor de pertinência 0,8 ou tem comprimento 3 com pertinência 0,3.

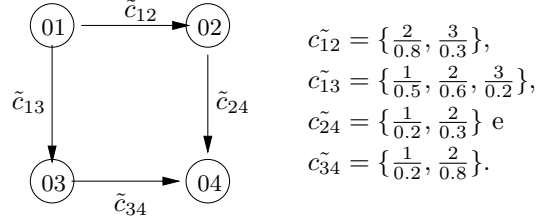


Figura 5.2: Exemplo apresentado para a aplicação dos algoritmos de Ford e o de Floyd em um grafo com parâmetros fuzzy

Para o segundo caso, apenas os trabalhos de (Delgado et al., 1990) e (Klein, 1991) apresentam teoria e propostas de resolução. Mesmo assim, os algoritmos são genéricos, isto é, foram desenvolvidos para qualquer problema em grafos e devem ser adaptados a cada problema.

Já o primeiro caso possui uma literatura significativa, sendo o trabalho de (Dubois e Prade, 1980) uma das primeiras referências. Este faz uma extensão de dois algoritmos clássicos do problema do caminho mínimo (algoritmos de Floyd e de Ford-Bellman) para a teoria fuzzy. No entanto, os resultados obtidos em ambos os estudos podem fornecer um comprimento sem um caminho associado. Veja o exemplo proposto em (Malik e Moderson, 2001):

Exemplo 5.2.1. (Malik e Moderson, 2001) *Seja uma rede fuzzy (Figura 5.2) tendo o nó 1 como origem e o nó 4 como destino. Usando programação dinâmica temos:*

$$\begin{aligned}
 \tilde{f}(4) &= \left\{ \frac{0}{1} \right\}, \\
 \tilde{f}(3) &= \{c_{34} + \tilde{f}(4)\}, \\
 &= \left\{ \frac{1}{0.2}, \frac{2}{0.8} \right\} + \left\{ \frac{0}{1} \right\} \\
 &= \left\{ \frac{1}{0.2}, \frac{2}{0.8} \right\}. \\
 \tilde{f}(2) &= \left\{ \frac{1}{0.2}, \frac{2}{0.3} \right\} \\
 \tilde{f}(1) &= \min\{c_{12} + \tilde{f}(2), c_{13} + \tilde{f}(3)\} \\
 &= \left\{ \frac{4}{0.2}, \frac{5}{0.3}, \frac{6}{0.3}, \frac{7}{0.2} \right\}
 \end{aligned}$$

O caminho 1 – 2 – 4 possui comprimento $\left\{ \frac{5}{0.2}, \frac{6}{0.3}, \frac{7}{0.3} \right\}$ e o caminho 1 – 3 – 4 têm comprimento $\left\{ \frac{4}{0.2}, \frac{5}{0.5}, \frac{6}{0.6}, \frac{7}{0.2} \right\}$. Portanto não existe caminho de comprimento $\left\{ \frac{4}{0.2}, \frac{5}{0.3}, \frac{6}{0.3}, \frac{7}{0.2} \right\}$ do nó 1 ao nó 4. Note também que nenhum caminho tem valor de pertinência 0.3 para o comprimento 5.

No caso, o problema do algoritmo é o uso dos operadores de mínimo fuzzy e de soma fuzzy (Capítulo 2).

Em (Klein, 1991) o problema verificado por Dubois e Prade pode ser contornado com o uso de dominância de conjuntos fuzzy ao invés do operador de mínimo fuzzy. Com isso determina-se o comprimento mínimo (geralmente igual a zero) e máximo que pode alcançar um caminho e trabalha-se com este valor em um algoritmo de programação dinâmica.

Um caso especial do problema do caminho mínimo fuzzy é tratado em (Okada e Gen, 1993) e (Okada e Gen, 1994), com o uso de intervalos no lugar de comprimento dos arcos e conceitos de ordenação parcial destes. Uma generalização do algoritmo de Dijkstra é utilizado em ambos trabalhos.

(Lin e Chern, 1993) propõem um algoritmo para o problema do caminho mínimo utilizando programação linear fuzzy, parametrização e partição do problema, mostrando como obter uma função de pertinência para esta solução.

Já (Chang e Lee, 1999) se baseiam na resolução do problema do caminho mínimo por programação dinâmica como em (Klein, 1991). O trabalho utiliza um método de ordenamento de número fuzzy (OERI) e utiliza métodos clássicos com base no valor resultante deste ordenamento. Até este ponto, os trabalhos visam encontrar um caminho com menor comprimento fuzzy.

Os trabalhos que tratam do conjunto fuzzy de soluções são mais recentes. Em (Okada e Soper, 2000) utiliza-se uma relação de ordenação de números fuzzy e resolve um problema de caminho mínimo multiobjetivo. Um método baseado em níveis de possibilidade de caminhos foi utilizado para reduzir o conjunto de soluções, formado por caminhos Pareto-ótimos, isto é, caminhos não dominados. Em (Chanas, 1987), alguns conceitos da teoria de possibilidade já são utilizados como um método para encontrar um único caminho mínimo com comprimento fuzzy. (Okada, 2001) desenvolveu um algoritmo, considerando o problema de caminho mínimo fuzzy (sendo um grafo crisp com parâmetros fuzzy), considerando os outros artigos estudados anteriormente mais o trabalho de (Yager, 1986), baseado na teoria de possibilidade. Introduz o conceitos de grau de possibilidade de um arco pertencer a um caminho mínimo. (Blue et al., 2002) é baseado no trabalho (Blue et al., 1997) onde é encontrado um valor de corte para limitar o número de caminhos a serem analisados. No trabalho mais recente, uma modificação do algoritmo dos k -caminhos mínimos (crisp) é utilizada para encontrar os caminhos com algum grau de possibilidade de ser o caminho mínimo, em tempo polinomial.

Neste capítulo, os algoritmo recentes do problema do caminho mínimo com parâmetros fuzzy serão analisados e um algoritmo modificado, baseado no estudo feito, será proposto. Ainda para o problema do caminho mínimo com parâmetros fuzzy será proposto um algoritmo genético para tentar tratar a complexidade do problema. Para o caso do problema do caminho mínimo em um grafo fuzzy, o algoritmo proposto em (Delgado et al., 1990) será adaptado ao problema e

algumas heurísticas serão propostas.

5.3 Caso I: grafo crisp e parâmetro fuzzy

Como foi visto na seção anterior, para este caso, a estrutura do problema é bem definida e o custo (\tilde{c}_{ij}) associado é fuzzy. Com isso, o comprimento de um caminho é um número fuzzy ($= \sum_{(i,j) \in P} \tilde{c}_{ij}$). Este tipo de problema é o que encontra o maior número de aplicações, visto que os parâmetros associados podem ser relacionados com grandezas *subjetivas*.

Os algoritmos estudados (ver Seção 5.2) apresentam formatos de solução diferentes. Nos artigos iniciais como em (Dubois e Prade, 1980), (Chanas, 1987) e (Klein, 1991) podemos observar que a solução deste problema foi proposta apenas como uma extensão do caso crisp, ou seja, apenas encontra um caminho (crisp) com comprimento fuzzy, tratado como caminho crítico (Bellman e Zadeh, 1970). Com isso, perde-se informações valiosas sobre caminhos *que poderiam ter um grau de pertinência ao conjunto de caminhos mínimos* (ver seção 4.1.2).

Os trabalhos mais recentes, (Okada e Soper, 2000), (Okada, 2001) e (Blue et al., 2002), apresentam como solução:

1. um conjunto fuzzy de caminhos, ou seja, caminhos crisps cada qual com um grau de *possibilidade* de ser o caminho mínimo;
2. um caminho mínimo fuzzy, ou seja, um subgrafo fuzzy onde cada arco tem um grau de possibilidade de pertencer ao caminho mínimo.

Para ambos os casos citados, o problema resultante é NP-completo. As próprias definições do grau de possibilidade do caminho e do arco determinam que todos os caminhos entre os nós de origem e destino sejam conhecidos para o cálculo do grau de possibilidade. (Okada, 2001) reconhece a limitação da abordagem neste ponto e sugere melhorias pelo desenvolvimento da teoria na parte de estrutura dos dados e técnicas de comparação.

Para ter uma solução completa do problema do caminho mínimo fuzzy, calcular o grau de possibilidade de cada arco/caminho é uma proposta atraente. O uso da teoria de possibilidade já foi proposta por (Chanas, 1987) mas apenas para encontrar o caminho crítico fuzzy. Já em (Okada, 2001) a teoria foi utilizada efetivamente para encontrar a solução. (Blue et al., 2002), embora não esteja explícito, também desenvolve trabalho neste mesmo campo. Em ambos os casos, foram observadas propriedades do problema que permitem cortes como, por exemplo, dos caminhos com grau de possibilidade nulo ou de caminhos com grau de possibilidade inferior a um determinado α -corte .

A seguir, vamos apresentar o algoritmo mais recente de (Okada, 2004) e propor alterações para melhorar a sua eficiência.

5.3.1 O estudo de Okada (2001 e 2004)

O algoritmo de (Okada, 2004) se baseia no método das múltiplas etiquetas, operações pelos conjuntos de α -níveis e pelos Teoremas 5.3.1 e 5.3.2 dados abaixo:

Teorema 5.3.1. *Se existe um caminho $p \in \mathcal{P}_{1r}$ tal que $D_p < \alpha$ sobre $G(r)$, então $D_{p \cup p'} < \alpha$ para qualquer $p' \in \mathcal{P}_{rn}$ sobre $G(N, A)$, onde $p \cup p'$ estabelece uma concatenação de caminhos.* \square

O Teorema 5.3.1 estabelece que, se durante a construção de um caminho for constatado que ele possui valor de possibilidade menor do que α , então o caminho do nó 1 até o nó n utilizando este trecho terá valor de possibilidade menor do que α . Veja o próximo teorema:

Teorema 5.3.2. *Para um dado $\alpha \in [0, 1]$, seja $A_1 = \{(i, j) \in A \mid D_{ij} \geq \alpha\}$ um conjunto de arcos sobre $G(N, A)$ e seja $A_2 = \{(i, j) \in A' \mid D_{ij} \geq \alpha\}$ um conjunto de arcos sobre $G = (N, A')$ onde $A' = A - \{(i, j) \mid D_{ij} < \alpha\}$. Então $A_1 = A_2$.* \square

Neste caso, o Teorema 5.3.2 mostra que se o objetivo é procurar caminhos com possibilidade maior do que α , então retirar os arcos que possuem grau de possibilidade estritamente menor do que α não interfere na busca.

No caso do método das múltiplas etiquetas, vamos definir alguns parâmetros. Seja um grafo $G = (N, A)$. Seja uma lista de etiquetas $[i, k]$ denotada L_r onde $[i, k]$ significa o nó precedente de r na k -ésima etiqueta na lista L_i . O caminho do nó 1 ao nó r pode ser composto voltando através das etiquetas das listas. O número de caminhos distintos do nó 1 ao nó r é dado por $|L_r|$. Seja o exemplo da Figura 5.3:

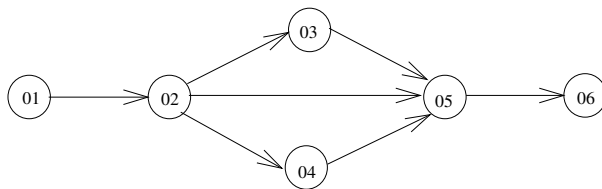


Figura 5.3: Exemplo da listas de etiquetas

A lista de etiquetas do nó 1 é igual a $L_1 = [-, -]$, pois 1 é o nó inicial. Para o nó 2 temos $L_2 = \{[1, 1]\}$, ou seja o nó 1 é o nó precedente do nó 2. Da mesma forma $L_3 = \{[2, 1]\}$ e $L_4 = \{[2, 1]\}$, o nó 2 é o nó precedente e a primeira etiqueta (e única) de L_2 foi selecionada para as listas L_3 e L_4 . Para o nó 5 temos $L_5 = \{[2, 1], [3, 1], [4, 1]\}$. Finalmente, para o nó 6 (destino): $L_6 = \{[5, 1], [5, 2], [5, 3]\}$, o nó 5 é o único nó precedente do nó 6, mas existem três etiquetas; a etiqueta $[5, 1]$ de L_6 indica a primeira etiqueta da lista L_5 , no caso $[2, 1]$, a etiqueta $[5, 2]$ se refere à segunda etiqueta da lista L_5 , $[3, 1]$ e a etiqueta $[5, 3]$ se refere a terceira etiqueta da lista L_5 , $[4, 1]$. Como $|L_6| = 3$, temos três caminhos possíveis entre o nó 1 e o nó 6. De fato, estes caminhos são: $1 \rightarrow 2 \rightarrow 5 \rightarrow 6$, $1 \rightarrow 2 \rightarrow 3 \rightarrow 5 \rightarrow 6$ e $1 \rightarrow 2 \rightarrow 4 \rightarrow 5 \rightarrow 6$. A composição destes caminhos pode ser feita utilizando as etiquetas das listas previamente determinadas. Por exemplo, tomando a primeira etiqueta de L_6 : $\{6, [5, 1], 5, [2, 1], 2, [1, 1], 1\}$ construímos o primeiro caminho. Os outros caminhos são construídos da mesma forma.

Portanto, o algoritmo das múltiplas etiquetas constrói todos os caminhos entre o nó origem e todos os outros nós. Comparando estes caminhos obtém-se o conjunto fuzzy solução. Mas o número de caminhos construídos e a comparação entre estes caminhos tornam o problema NP-completo. Para isso, Okada (2004) enunciou os teoremas e propôs a utilização do conceito de α -níveis. Para um determinado α -nível, se um caminho de um nó 1 a um nó r , $1 < r < m$, tiver valor de possibilidade menor do que α , então a etiqueta referente a este caminho é retirado (Teorema 5.3.1). O Teorema 5.3.2 garante que a retirada dos arcos não deve interferir na busca pelos caminhos com valor de possibilidades maiores do que α . Esta combinação de conceitos garante a redução no número de caminhos a serem construídos e, portanto, comparados. O algoritmo proposto por (Okada, 2004) é descrito em dois procedimentos dados a seguir.

5.3.2 O algoritmo de Okada (2001 e 2004): CM-OK04

Procedimento 01:

Passo 0: (*Definir parâmetros*) Estabeleça um número de partições $h \in \mathbb{Z}^+$ que dividirá o intervalo $[0, 1]$.

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

$$A' \leftarrow A.$$

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

Passo 2: Execute o procedimento 02 com o parâmetro α .

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

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

O procedimento 1 é responsável por definir o α -nível (Passo 0) e eliminar os arcos do subgrafo utilizado na próxima iteração (passo 3), utilizando o resultado do Teorema 5.3.2. Vejamos o procedimento 2:

Procedimento 02:

Passo 0: Para o parâmetro α , atualize $(\tilde{l}_{ij})_\alpha$ ($\forall (i, j) \in A'$).

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

Passo 2: Para $r = 2, \dots, n$ repita os passo 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 têm somente um nó precedente), então retorne ao passo 2. Senão vá para o passo 5.

Passo 5: Para cada par de etiquetas $[i_s, k_s]$ e $[i_t, k_t] \in L_r$, construa os caminhos p_s e p_t respectivamente utilizando as listas até chegar ao 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 01.

No procedimento 2, o passo 3 constrói a lista de um nó i concatenando as listas dos nós L_j precedentes ao nó i , colocando j no campo do nó precedente ao nó i e colocando no segundo campo a posição daquela etiqueta na lista L_j . No passo 5, é utilizado o Teorema 5.3.1, em que se o subcaminho tiver possibilidade menor do que α esta etiqueta é descartada, pois o caminho do nó 1 ao nó n terá possibilidade menor do que α . Com isso, o número de comparações diminui.

No trabalho de (Okada, 2001) é constatado que o passo 5 do procedimento 2, que trata da construção e comparação de caminhos precisa ser mais estudado, pois é o passo que consome quase todo o tempo computacional.

5.3.3 O estudo de Blue *et al.* (2002)

Em (Blue et al., 1997) (e (Blue et al., 2002)) é realizada uma taxonomia sobre a incerteza em um grafo e propostos alguns algoritmos para resolver o problema em grafos fuzzy. No caso dos algoritmos, a idéia principal é a determinação de um valor de corte para o descarte dos comprimentos fuzzy que possuem valor de pertinência nulo ao conjunto solução (veja Capítulo 4). O cálculo do corte κ é realizado considerando caminhos não-interativos, isto é, todos os arcos são considerados no cálculo, diferente do trabalho de (Okada, 2004) em que os arcos em comum entre dois caminhos são retirados dos cálculos. Dado o grafo crisp G com comprimentos iguais a $c_{ij} = \sup\{supp\{\tilde{c}_{ij}\}\}$, $(i, j) \in A$, temos que o valor do corte é dado pela resolução do problema crisp:

$$\kappa = \min_{p \in \mathcal{P}_{1n}} \{l_p\} = \min_{p \in \mathcal{P}_{1n}} \left\{ \sum_{(i,j) \in p} c_{ij} \right\}$$

onde p é um caminho pertencente ao conjunto de caminhos do nó 1 ao nó n , \mathcal{P}_{1n} . Como o trabalho não utiliza a teoria de possibilidade, foi elaborado uma proposição que demonstra que este corte é válido para esta dissertação (Proposição 2.7.2, Capítulo 2). Os autores também citam que o conjunto de caminho interativos não nulos é um subconjunto próprio dos caminhos não-interativos não nulos na maioria das vezes, o algoritmo faria as comparações praticamente com os caminhos que vão participar da solução.

Um exemplo do corte proposto é dado na Figura 5.4. Os números fuzzy trapezoidais são avaliados em seus valores ínfimos e supremos do conjunto suporte $supp\{\tilde{c}_{ij}\}$. O menor supremo é tomado como valor de corte κ . Qualquer número fuzzy em que o ínfimo seja maior do que κ pode ser descartado. Para o exemplo, o supremo do número fuzzy (a) foi escolhido e, dado este valor para κ , o número fuzzy (c) deve ser retirado do conjunto solução.

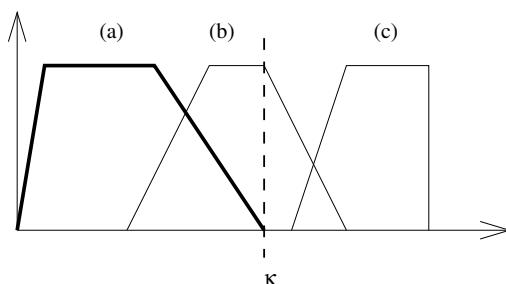


Figura 5.4: Comparação dos comprimentos fuzzy

Em (Blue et al., 1997) sugere-se procurar um algoritmo adequado para gerar os caminhos e aplicar o teste de corte antes de efetuar a comparação dos caminhos para determinação dos valores de pertinência. Em Blue et al. (2002), cita-se o uso de um algoritmo modificado do

algoritmo proposto em (Eppstein, 1994), mas a modificação sugerida não é apresentada no trabalho. O algoritmo de (Eppstein, 1994) deve encontrar os k -caminhos mínimos em tempo polinomial.

5.3.4 O estudo de Okada e Soper, 2000

Neste estudo, o conjunto solução procurado é o dos caminhos *não dominados*, isto é, Pareto-ótimas. A seguinte asserção define o conceito de dominância fuzzy:

Asserção 5.3.1. (Okada e Soper, 2000) Sejam $\tilde{a} = (\underline{a}, a, \bar{a})$ e $\tilde{b} = (\underline{b}, b, \bar{b})$ dois números fuzzy. Então $\tilde{a} \succsim \tilde{b}$ (\tilde{a} domina \tilde{b}) se, e somente se, as seguintes desigualdades ocorrerem: $\underline{a} \leq \underline{b}$, $a \leq b$ e $\bar{a} \leq \bar{b}$.

Seja o exemplo da Figura 5.5. Seja $\tilde{a} = (\underline{a}, a, \bar{a})$ o número fuzzy (a) da figura, $\tilde{b} = (\underline{b}, b, \bar{b})$ o número fuzzy (b) e $\tilde{c} = (\underline{c}, c, \bar{c})$ o número fuzzy (c). Notamos que $\tilde{b} \not\prec \tilde{c}$ pois $\underline{b} \leq \underline{c}$, $b \leq c$ mas

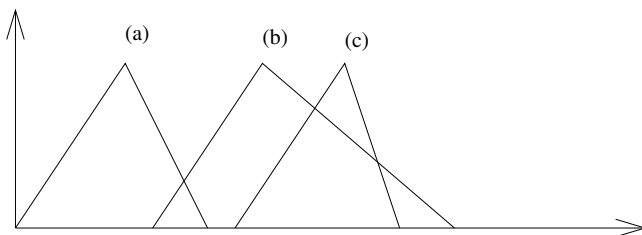


Figura 5.5: Exemplo de dominância de números fuzzy

$\bar{b} > \bar{c}$. No entanto, \tilde{a} domina tanto \tilde{b} quanto \tilde{c} pela Asserção 5.3.1. Para compreender a utilização do conceito de dominância para a construção do conjunto fuzzy de soluções associa-se \underline{a} a uma solução crisp otimista, \bar{a} a uma solução crisp pessimista e a a uma solução neutra ou indiferente ((Lin e Chern, 1993) e (Chang e Lee, 1999)). Se também adotarmos tais conceitos, podemos pensar que uma solução que *domina* outra solução por ter melhor resultado em qualquer um dos problemas otimista, pessimista ou indiferente.

Este estudo também considera o grau de possibilidade como nos estudos anteriores das Seções 5.3.1 e 5.3.3. O algoritmo proposto também considera os α -níveis e utiliza o método das múltiplas etiquetas como em (Okada, 2004), mas a etiqueta contém mais informações, devido ao teste de dominância. A lista neste algoritmo é dado por:

$$L_r = \{[\tilde{l}, (i, k)]_j\} \quad (5.7)$$

onde \tilde{l} é o comprimento do subcaminho do nó 1 ao nó r , i é o nó precedente ao nó r ; k é a posição da etiqueta correspondente na lista L_i , e j é a identificação da etiqueta para o nó r .

Para apresentar o algoritmo, o conceito de ordem lexicográfica precisa ser definida:

Definição 5.3.1. Segundo (Okada e Soper, 2000), sejam $\tilde{a} = (\underline{a}, a, \bar{a})$ e $\tilde{b} = (\underline{b}, b, \bar{b})$ dois números fuzzy. Então \tilde{a} é lexicograficamente menor do que \tilde{b} se um dos seguintes casos for verdadeiro:

$$\begin{aligned} \underline{a} &< \underline{b} \\ \underline{a} &= \underline{b} \text{ e } a < b \\ \underline{a} &= \underline{b} \text{ e } a = b \text{ e } \bar{a} < \bar{b} \end{aligned}$$

Com isso, temos os conceitos básicos para compreender o algoritmo proposto por (Okada e Soper, 2000), dado na seção a seguir.

5.3.5 O algoritmo de Okada e Soper, 2000: CM-OS00

Passo 0: Associe a etiqueta $[\tilde{0}, (-, -)]_1$ ao nó 1. Faça o ponteiro associado a esta etiqueta pertencer ao conjunto temporário, $T \leftarrow (1, 1)$, e associe o conjunto vazio ao conjunto permanente, $P \leftarrow \emptyset$.

Passo 1:(*Seleção das etiquetas*)

- (i) Se $T = \emptyset$, vá para o Passo 4. Caso contrário, entre todos os ponteiros temporários, determine o lexicograficamente menor. Seja esta a k -ésima etiqueta associada com o nó i .
- (ii) Retire o ponteiro do conjunto temporário e associe-o ao conjunto permanente

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

Passo 2: (*procure novas etiquetas e teste a dominância*) Para cada nó $j \in N$ tal que $(i, j) \in A$, execute os passos (i), (ii) e (iii)

- (i) Determine as distâncias fuzzy:

$$\tilde{d}_l(p_{1j}) = \tilde{d}_k(p_{1i}) \oplus \tilde{l}_{ij}$$

onde $\tilde{d}_l(p_{1j})$ é a distância do nó 1 ao nó j , segunda a etiqueta l , e \tilde{l}_{ij} é o comprimento do arco (i, j) . \oplus é uma soma fuzzy.

- (ii) Seja $[\tilde{d}_l(p_{1i}), (i, k)]_l$ a nova etiqueta, faça $T \leftarrow T \cup \{(j, l)\}$
- (iii) Confira a dominância de todas as etiquetas temporárias do nó j
- Para cada $\tilde{d}_u(p_{1j})$ tal que $(j, u) \in T \cup P$, se $\tilde{d}_l(p_{1j}) \succ \tilde{d}_u(p_{1j})$, descarte a l -ésima etiqueta do nó j e o ponteiro correspondente de T e encerre a parte (iii).
 - Para cada $\tilde{d}_u(p_{1j})$ tal que $(j, u) \in T$, se $\tilde{d}_l(p_{1j}) \prec \tilde{d}_u(p_{1j})$, descarte a u -ésima etiqueta e o ponteiro correspondente (j, u) de T .
 - Caso contrário, não há nada a ser feito.

Passo 3: Volte ao Passo 1.

Passo 4: Encontre os caminhos não dominados do nó 1 ao nó m . FIM.

5.3.6 Uma proposta de melhoria no algoritmo de Okada

Utilizando algumas propriedades de (Okada e Soper, 2000) e (Blue et al., 2002), espera-se melhorar a eficiência do algoritmo de (Okada, 2001) com as seguintes modificações:

1. Utilizando o corte proposto por (Blue et al., 2002), os caminhos com valor de possibilidade nulo podem ser eliminados sem a necessidade de construção deste para comparação. O Passo 3 do procedimento 2 do algoritmo de Okada pode ser alterado, incluindo este teste.
2. Guardando o grau de possibilidade de um caminho ser menor que o outro junto a etiqueta elimina a necessidade do cálculo em cada α -corte. O valor do grau de possibilidade de cada arco também será preciso, não aproximado como em (Okada, 2001).
3. Para que as propostas de melhoria dos itens anteriores se tornem efetivas, um modelo de etiqueta, $[(i, k), \tilde{l}, w]$, semelhante ao utilizado em (Okada e Soper, 2000) é introduzido no algoritmo, sendo:

(i, k) : i é o nó predecessor de r e k é a posição desta etiqueta na lista L_i .

\tilde{l} : comprimento fuzzy do caminho P entre o nó 1 e o nó r dado por esta etiqueta.

w : grau de possibilidade do caminho dado por esta etiqueta ser o caminho mínimo entre o nó 1 e o nó r .

4. A comparação feita no passo 5 do procedimento 2 pode ser feita de modo a reduzir pelo menos à metade os cálculos realizados até o momento, observando os valores modais (Proposição 2.7.1, Capítulo 2). Isto também pode incentivar cortes de um caminho antes que este faça muitas comparações com outros caminhos.

Com base nas afirmações feitas, propomos um algoritmo modificado, baseado no algoritmo de Okada, descrito a seguir:

5.3.7 Algoritmo modificado (CM-OK04MOD):

Passo 0: (*Preparar os parâmetros*) Calcule $\kappa = \min_{P \in \tilde{G}}(l_{\tilde{P}})$, $\kappa \in \mathfrak{R}^n$, sendo k_i o valor de corte em cada nó i . Seja α dado.

Passo 1: (*Inicializar*) Faça: $L_1 = \{[(-, -), \tilde{0}, 1]\}$

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

Passo 3: (*Gerar etiquetas e eliminar caminhos com $D_p = 0$*) Para $j \in B_r$, faça:

Seja $L'_j \leftarrow L_j$. Para cada i em $[(i, k), \tilde{l}, w] \in L'_j$, faça

- Atualize $\tilde{l} \leftarrow \tilde{l} \oplus a_{ij}$.
- Se $\inf\{\text{supp}\{\tilde{l}\}\} > k_j$, faça $L'_j \leftarrow L'_j - \{[(i, k), \tilde{l}, w]\}$.
Caso contrário, substitua i por j na etiqueta

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

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

Passo 5: (*Verificar D_p em caminhos iterativos*) Ordene as etiquetas em ordem não decrescente do valor modal de \tilde{l} . Para $s = 1$ até $|L_j| - 1$, faça:

- construa o caminho P_s por $[(i_s, k_s), \tilde{l}_s, w_s]$
- Para $t = |L_j|$ até 2, faça:
 - componha o caminho P_t de $[(i_t, k_t), \tilde{l}_t, w_t]$
 - Calcule $\bar{w} = \text{Poss}(\sum_{(i,j) \in p_s} l_{ij} \leq \sum_{(i,j) \in p_t} l_{ij})$
 - Se $\bar{w} = 0$ ou $\bar{w} < \alpha$ então $L_r \leftarrow L_r - [(i_s, k_s), \tilde{l}_s, w_s]$.
Caso contrário, $w_s \leftarrow \min\{w_s, \bar{w}\}$.

Passo 6: (*Cálculo do grau de possibilidade de cada arco*) Seja L_n e $D_{ij} = 0$, $(i, j) \in A$. Para cada etiqueta $[(i_s, k_s), \tilde{l}_s, w_s]$, faça:

- Reconstrua o caminho P_s
- Para cada arco (i, j) de P_s : $D_{ij} = \max\{D_{ij}, w_s\}$.

Retorne D.

Observação: O corte feito pelo α -corte foi mantido para o caso em que o decisor preferir uma solução em que somente os arcos com possibilidade maior do que α participem.

5.3.8 Complexidade do algoritmo proposto

Como no algoritmo de Okada, deve existir apenas um arco entre dois nós e que para um arco $(i, j), i < j$, o grafo com maior número de arcos é dado na Figura 5.6:

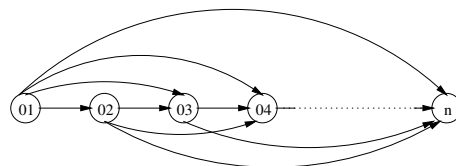


Figura 5.6: Grafo máximo para o algoritmo de Okada modificado

em um total de $\frac{m(m-1)}{2}$ arcos para m nós. Considerando o pior caso, em que todos os caminhos possuem possibilidade não nula, temos:

$$\begin{aligned}
 i = 2 & \rightarrow P_2 = 1 \\
 i = 3 & \rightarrow P_3 = 2 \\
 i = 4 & \rightarrow P_4 = (P_1 + P_2) + P_3 = P_3 + P_3 = 4 \\
 i = 5 & \rightarrow P_5 = (P_1 + P_2 + P_3) + P_4 = P_4 + P_4 = 8 \\
 & \vdots \\
 i = m & \rightarrow P_m = 2 \times P_{m-1} = 2^{m-2}
 \end{aligned}$$

definindo $P_1 = 1$. Portanto, um problema com m nós pode ter até 2^{m-2} caminhos. O número de comparações por nó i é igual a $P_i(P_i - 1)$ pois temos que comparar um caminho com todos os outros. Então, o número de comparações que devem ser realizadas do nó 2 ao nó n é dada pela função:

$$F = \sum_{i=2}^n P_i(P_i - 1) = \sum_{i=0}^{n-2} 2^i(2^i - 1)$$

O número de operações realizadas a cada comparação será feita F vezes, desconsiderando ainda o tempo que é utilizado para montar os subcaminhos para comparação. Isto seria feito em cada α -corte pelo método de (Okada, 2001).

No caso do algoritmo modificado, para o caso mais complexo, a questão combinatorial também não é contornada, mesmo que o número de comparações seja no máximo metade da

realizada por (Okada, 2001). A complexidade é verificada nos exemplos dados a seguir devido ao grau de incerteza dos dados e ao tipo de problema.

Para calcular o número de caminhos e de comparações para problemas gerais sejam:

B_i : lista dos nós que antecedem o nó i .

p_j : número de caminhos do nó 1 ao nó j .

r_j : número de comparações realizadas entre os caminhos do nó 1 ao nó j .

O cálculo do número de caminhos pode ser feito da seguinte forma:

$$p_j = \sum_{i \in B_j} p_i, \quad p_1 = p_2 = 1, \quad i = 2, \dots, n$$

com p_j sendo o maior número de caminhos que podem ser encontrados, desconsiderando os cortes. O número de comparações é baseado no número de caminhos, na equação dada abaixo:

$$r_j = p_j \times (p_j - 1)$$

sendo r_j o número máximo de comparações, também desconsiderando a ordenação e os cortes propostos.

5.3.9 Proposta de um algoritmo genético (CM-AG) para o problema de caminho mínimo com parâmetro fuzzy

A motivação para a proposta de um algoritmo genético reside na complexidade dos algoritmos estudados nas seções anteriores. A dimensão e a incerteza do problema interferem no tamanho do conjunto fuzzy de soluções (Okada e Soper, 2000). Nestes casos, um algoritmo genético que encontre um *bom* subconjunto solução é uma opção atraente.

Os algoritmos para o problema clássico de caminho mínimo são eficientes e assim, não incentivam o uso de heurísticas na resolução. Com isso, não foram encontrados heurísticas ou meta-heurísticas que inspirassem a construção do algoritmo genético. Em um algoritmo genético, a escolha da codificação é importante para garantir a boa representação de um candidato a solução factível do problema. Outros fatores como os operadores e os parâmetros também devem ser bem estudados. Aqui vamos apresentar uma primeira sugestão de algoritmo genético para tratar este problema. Para facilitar a compreensão desta seção, algumas definições e conceitos necessários sobre algoritmos genéticos são apresentados no Apêndice B.

Codificação: A codificação utilizada retrata o caminho de forma mais direta: cada bit é do

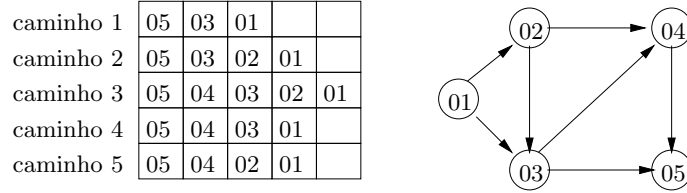


Figura 5.7: Exemplo de codificação para o problema do caminho mínimo fuzzy

nó visitado sendo que os arcos serão dados por $(bit(i + 1) \rightarrow bit(i), i = 1, \dots, r - 1)$, com r o número de arcos que compõe o caminho ($r = 1, \dots, m, m = |N|$). Por exemplo, na Figura 5.7, o caminho 1 contém os arcos $\{(1, 3), (3, 5)\}$, o caminho 2 contém os arcos $\{(1, 2), (2, 3), (3, 5)\}$ e assim por diante.

Recombinação (crossover): é feito pela união das informações (nós visitados) contidas nos dois cromossomos escolhidos (caminhos-pais). Com isso são escolhidos os nós que farão parte do filho somente dentre os nós pertencentes a esta união.

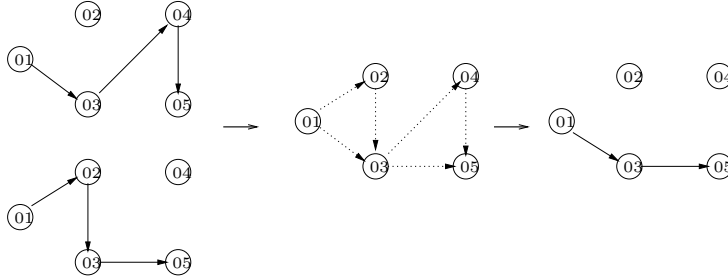


Figura 5.8: Exemplo de crossover para o problema do caminho mínimo fuzzy

Mutação: neste procedimento, um nó do caminho é proibido e um novo trecho será construído usando preferencialmente arcos que fazem parte do subcaminho a ser descartado.

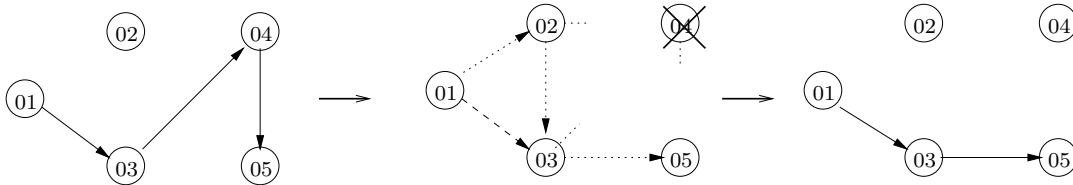


Figura 5.9: Exemplo de mutação para o problema do caminho mínimo fuzzy

Inicialização: Parte da população inicial é dada como uma mutação do melhor caminho encontrado utilizando o caminho mínimo do grafo crisp G^* associado ao grafo fuzzy proposto,

com custo dos arcos igual ao ínfimo valor modal dos custos fuzzy. O restante é construído escolhendo, de forma aleatória, os nós predecessores até chegar ao nó 1.

Seleção: A nova população é formada pelos filhos gerados durante a recombinação e a mutação, uma parte dos melhores indivíduos da população anterior e o restante da nova população será escolhido aleatoriamente.

Fitness: A medida de avaliação calcula a possibilidade do caminho ser menor que o melhor caminho encontrado em um procedimento realizado previamente no algoritmo genético.

O algoritmo genético segue o esquema apresentado no Apêndice B.

5.3.10 Exemplos numéricos

Vamos utilizar o exemplo dado em (Okada e Soper, 2000) em que deve ser encontrado um caminho entre Boston e Los Angeles, usando como base o mapa de tempos de percurso entre as cidades (Apêndice C, Tabela C.2). O número fuzzy associado será trapezoidal com 20% de platô e 10% de desvio. Outros exemplos serão de uma rede óptica italiana (italia1)(Ali et al., 2000) (Apêndice C, Tabela C.1) e o exemplo descrito na Figura 5.10, CMex. A cada arco da Figura 5.10, um número triangular fuzzy está associado.

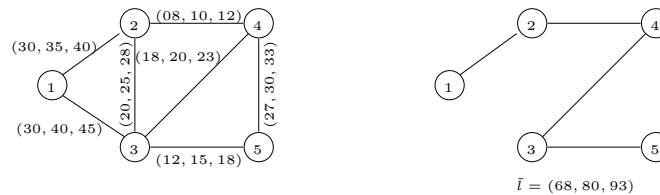


Figura 5.10: Exemplo de uma rede fuzzy:CMex

As comparações serão feitas entre o algoritmo implementado por (Okada, 2001) e o algoritmo modificado (alg-M) proposto. A comparação com o trabalho de (Okada e Soper, 2000) não será feita pois este algoritmo encontra os caminhos não dominados e não todo o conjunto fuzzy.

Como nos algoritmos das Seções 5.3.2 e 5.3.7 o ponto crítico é o número de comparações realizadas entre os subcaminhos e entre os caminhos, estes dados serão comparados.

Nota-se na Tabela 5.1 que, para casos em que poucos caminhos participam da solução do problema, as modificações propostas diminuem consideravelmente o número de comparações e, em conseqüência, o tempo computacional. Para problemas em que muitos caminhos participam

exemplo	CM-OK04	tempo (s)	CM-OK04MOD	tempo (s)
CMex	127	0,8900	14	0,0400
italia1	2414	1,8820	658	0,4910

Tabela 5.1: Número de comparações realizadas nos testes com os exemplos CMex e italia1

da solução (vide Tabela 5.2), apenas a modificação proposta referente a ordenação das etiquetas e a comparação ordenada destes surte efeito, reduzindo em cerca de 50% o número de comparações. Observando o número de comparações realizadas pelo algoritmo de (Okada, 2001) e o algoritmo modificado, CM-OK04MOD, vide Tabela 5.2, nota-se que o número de comparações cresce à medida que os caminhos vão sendo construídos. Mas este valor ainda está muito abaixo do número de comparações que deveriam ser feitas entre 42.167.110 caminhos possíveis entre o nó 1 e o nó 70 com até 2.5225×10^{15} comparações durante o processo (1.55×10^{15} só em comparações entre os caminhos do nó 1 ao nó 70, o restantes são comparações entre os subcaminhos dos outros nós), para $\alpha = 0$.

m	Máximo	CM-OK04	%	CM-OK04MOD	%
10	30	6	20	3	10
20	7108	324	4,56	54	0,76
30	$1,51 \times 10^6$	6.145	0,41	508	$3,36 \times 10^{-3}$
40	$1,17 \times 10^8$	26.846	$2,83 \times 10^{-4}$	3.307	$2,83 \times 10^{-4}$
50	$1,83 \times 10^{10}$	141.205	$7,72 \times 10^{-5}$	13.631	$7,45 \times 10^{-5}$
60	$1,66 \times 10^{12}$	660.520	$3,98 \times 10^{-5}$	62.702	$3,78 \times 10^{-6}$
70	$2,52 \times 10^{15}$	10.264.377	$4,07 \times 10^{-7}$	677.037	$2,69 \times 10^{-8}$

Tabela 5.2: Comparações feitas no exemplo da rede americana

No exemplo da rede americana, 653 caminhos possuem o grau de possibilidade máximo ($Poss = 1$) sendo realizadas 1.097.559 comparações no algoritmo modificado (utilizando $\alpha = 1$ como valor de corte) para encontrá-los. Isso é resultado da incerteza dos valores associados. Em (Okada e Soper, 2000) já foi estudado a influência da incerteza dos comprimentos no número de caminhos que participam da solução. Para o caso de caminhos não-dominados, (Okada e Soper, 2000) fornece oito caminhos, sendo que apenas um com possibilidade igual a 1 é apresentado como solução para o problema da rede americana. Para o decisor, uma *filtragem* da solução pode ser muito útil, neste caso.

Para este mesmo problema, com tempo descrito por números triangulares fuzzy, ou seja, sem *flat* e com desvio de 10%, o número de comparações realizadas diminui drasticamente para

2.672.536 comparações para o algoritmo de Okada e 1.093.446 comparações no algoritmo modificado. Uma amostra da dependência do tamanho do conjunto solução sob a incerteza dos dados.

O algoritmo genético procura gerar um subconjunto relevante de soluções factíveis do problema, uma vez em que ele pode apresentar dificuldades em encontrar todo o conjunto solução. Para ilustrar, seja o problema da rede rodoviária americana. Esta rede possui 42.167.110 caminhos possíveis entre os nós 1 e 70 com um máximo de $2,52 \times 10^{15}$ comparações necessárias pelo algoritmo exato. O conjunto solução, para custos representados por números fuzzy triangulares com espalhamento de 10% em relação ao valor modal, possui 632 caminhos. O algoritmo genético foi executado por cinco vezes e em nenhum caso se encontrou todo o conjunto solução. Apesar da população manter sua diversidade, a dificuldade de encontrar caminhos com valor de possibilidade baixo interfere na construção do conjunto solução. Em compensação, os caminhos com alto grau de possibilidade são fáceis de encontrar. Para este problema, os caminhos com grau de possibilidade acima de 0,6 foram encontrados corretamente. Observando os valores de possibilidade dos arcos, os caminhos com grau de possibilidade abaixo de 0,2 são aqueles mais difíceis de obter.

Apesar das dificuldades mencionadas, o algoritmo genético é atraente uma vez que os caminhos com maior grau de possibilidade são obtidos em tempo computacional reduzido. No exemplo da rede rodoviária americana, o resultado foi obtido em aproximadamente 120 segundos, que é muito menor do que o observado no algoritmo modificado (alg-M), 3157 segundos, em média. Portanto, estamos tentando alguma sofisticação na questão da diversidade da população inicial. Outra melhoria na aproximação do algoritmo é no cálculo do fitness. Será considerado um *subconjunto* dos melhores caminhos fuzzy encontrados para a comparação ao invés do caminho de grau máximo. Assim, pretende-se obter valores de possibilidade mais próximos aos encontrados pelo algoritmo exato.

5.4 Caso II: o problema do caminho mínimo em um grafo fuzzy

Neste caso a estrutura do problema não é bem definida. Para cada nó e para cada arco estarão associadas funções de pertinência $\sigma : N \rightarrow [0, 1]$ e $\mu : N \times N \rightarrow [0, 1]$ respectivamente, sendo que:

$$\mu(i, j) \leq \sigma(i) \wedge \sigma(j)$$

ou seja, o grau de pertinência do arco (i, j) não pode ser maior que o grau de pertinência dos respectivos nós. Para cada arco temos associado um valor de pertinência e um custo, que

também pode ser um número fuzzy.

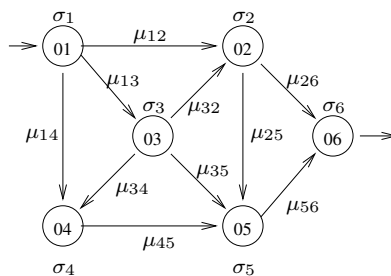


Figura 5.11: Exemplo de um problema de caminho mínimo fuzzy

O caminho entre o nó 1 e o nó m terá um comprimento l_{1m} e um valor de pertinência associado igual a do arco de menor pertinência (Rosenfeld, 1975), ou seja:

$$\mu_{P_{1m}} = \bigwedge_{(i,j) \in P_{1m}} \mu_{ij} = \min_{(i,j) \in P_{1m}} \mu_{ij}$$

sendo P_{1m} um caminho do nó origem 1 ao nó destino m e $\mu_{P_{1m}}$ o seu valor de pertinência. Como de $\mu_{P_{1m}}$ em um grafo com estrutura fuzzy depende de seu arco mais fraco, podemos pensar em resolver este problema utilizando α -cortes. Para simplificar o problema, vamos considerar que o custo dos arcos são valores crisp.

A idéia de utilizar os α -cortes para construir o conjunto solução foi proposta inicialmente por (Delgado et al., 1990) para um algoritmo fundamental que fosse utilizado para qualquer problema em grafo fuzzy. Na próxima seção ele será adaptado ao problema de caminho mínimo e analisado.

5.4.1 Algoritmo proposto por Delgado et. al, 1990

Aqui, a idéia do trabalho (Delgado et al., 1990) será apresentada adaptada ao problema do caminho mínimo. As propriedades que garantem a otimalidade do problema estão apresentadas no trabalho citado.

Seja o grafo suporte $G^* = (N^*, A^*)$ associado ao grafo fuzzy $\tilde{G} = (\sigma, \mu)$ (isto é, cujos arcos tenham graus de pertinência estritamente maiores do que zero: $\sigma_i > 0$ e $\mu_{ij} > 0$, $i, j \in N$ e $(i, j) \in A$). Resolvendo o problema em G^* , obtemos um caminho mínimo P_{α_0} , com um valor de pertinência α_0 , definido pelo seu arco mais fraco. Este caminho possui o menor comprimento possível tanto no grafo fuzzy \tilde{G} quanto no grafo crisp associado G_{α_0} . Para obter um caminho

com maior grau de pertinência, podemos considerar o grafo crisp resultante do conjunto de corte abaixo:

$$A_{\alpha_1} = \{(i, j) \in A \mid \mu_{ij} > \alpha_0\} \quad (5.8)$$

onde A_{α_1} é um conjunto de arcos crisp. Então, o grafo crisp $G_{\alpha_1} : (N, A_{\alpha_1})$ associado pode ser resolvido por meio de algoritmos clássicos para obter um caminho, P_{α_1} , com grau de pertinência maior do que P_{α_0} , mas também com comprimento maior. Este procedimento gera um conjunto de caminhos $\mathcal{P} = \{\frac{P_{\alpha_0}}{\alpha_0}, \frac{P_{\alpha_1}}{\alpha_1}, \dots, \frac{P_{\alpha_q}}{\alpha_q}\}$, sendo o caminho P_{α_i} associado ao seu valor de pertinência α_i , $i = 1, \dots, q$ e q a última iteração do algoritmo que possui um caminho entre o nó origem e o nó destino.

Com base na observação acima, podemos construir o conjunto solução calculando o caminho mínimo para cada conjunto de corte determinado pelo grau de pertinência do caminho mínimo do nível anterior. Abaixo apresentamos o algoritmo adaptado de DVV90CM:

Algoritmo para caminho mínimo fuzzy: DVV90CM

Passo 0: Seja um grafo fuzzy $G = (\sigma, \mu)$ com custo c_{ij} , $(i, j) \in A$. Seja $k = 0$; $\alpha_0 = 0$. Seja o conjunto suporte $G^* = \{(i, j) \in A \mid \mu_{ij} > 0\}$.

Passo 1: Resolva o problema em G^* e seja P o caminho mínimo. Faça:

$$\begin{aligned} P_k &\leftarrow P \\ l_k &\leftarrow \sum_{(i,j) \in P} c_{ij} \\ \alpha_k &\leftarrow \bigwedge_{(i,j) \in P} \mu_{ij} = \min_{(i,j) \in P} \{\mu_{ij}\} \end{aligned}$$

onde α_k é o valor de pertinência do caminho. Vá ao Passo 2.

Passo 2: Retire os arcos com valor de pertinência $\leq \alpha_k$ do grafo. Faça $k \leftarrow k + 1$. Se o grafo apresentar componentes não conectados \rightarrow FIM. Caso contrário volte ao Passo 1.

5.4.2 Complexidade do algoritmo

Com o algoritmo proposto acima, a complexidade do algoritmo dependerá do algoritmo crisp escolhido para a resolução. Considere um grafo fuzzy com m nós e n arcos. Como o algoritmo acima faz uso dos α -cortes, no pior caso serão necessários $n - m + 1$ iterações do algoritmo. A ordem de complexidade do algoritmo será de $O((n - m)O_k)$, onde O_k é a ordem de complexidade do algoritmo crisp para o grafo resultante utilizado na iteração k .

5.4.3 Exemplo Numérico

Seja o exemplo dado na Figura 5.12:

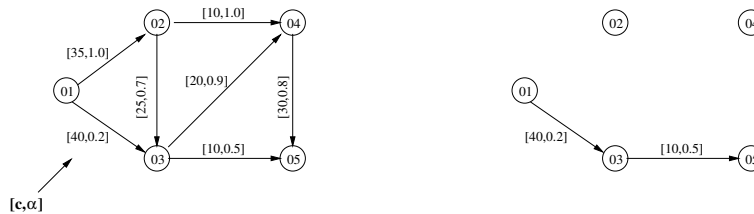


Figura 5.12: Exemplo de um grafo fuzzy e primeira solução

onde c é o custo do arco e α é o valor de pertinência do arco. Para o conjunto suporte temos a solução dada. Este caminho mínimo possui comprimento igual a 50 e grau de pertinência igual a 0,2. Então descartamos os arcos com pertinência menor ou igual a 0,2. Com o novo subgrafo encontramos uma nova solução, procedemos um novo corte até o subgrafo ser desconexo (Figuras 5.13, 5.14 e 5.15):



Figura 5.13: Comprimento = 70, $\alpha = 0,5$



Figura 5.14: Comprimento = 75, $\alpha = 0,8$

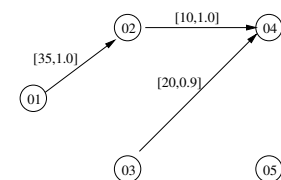


Figura 5.15: Grafo Desconexo \Rightarrow FIM

5.4.4 Algumas heurísticas para o problema do caminho mínimo em um grafo fuzzy

No caso de um grafo denso ou completo e com grande variação nos valores de pertinência dos arcos, o procedimento pode ser muito caro em termos computacionais e o conjunto de soluções pode ser muito grande e pouco útil ao decisor. Propomos algumas formas de contornar este problema, encontrando apenas um determinado caminho, ou um subconjunto de caminhos:

Heurística 1: *Resolver para um valor de confiabilidade escolhido pelo usuário (H_1):* Escolher um grau de pertinência e resolver o problema, encontrando apenas um caminho como solução (Passo 1 do algoritmo da Seção 5.4.1).

Heurística 2: *Redução do número de intervalos (H_2):* O usuário pode determinar o número máximo de partições para o α -corte. Sendo assim, o passo 1 teria a seguinte alteração

$$\alpha_k \leftarrow \max\{\min\{k * \delta, 1\}, \alpha_{k-1}\}$$

com $\delta = \frac{1}{p}$ e $p > 0$ o número de partições do intervalo. Observando sempre o maior valor entre a próxima iteração e a iteração atual garante que não sejam feitos cálculos desnecessários em intervalos que não possuem caminhos diferentes dos já calculados.

Heurística 3: *Solução com melhor relação (valor de pertinência) \times (comprimento do caminho) (H_3):* Utilizar um procedimento proposto por (Chunde, 1996) para o problema da árvore geradora mínima onde ele balanceia a pertinência da árvore proposta e o seu custo total. O problema seria modelado como:

$$\begin{aligned} \text{Min} \quad & g(P) \\ \text{s.a.} \quad & g(P) = F(P)c(P) \\ & F(P) = \bigvee_{(i,j) \in P} (1 - \mu_{ij}) \\ & c(P) = \sum_{(i,j) \in P} c_{ij} \\ & P \in \mathcal{P}(\mathcal{N}) \end{aligned} \tag{5.9}$$

onde $F(P) = \bigvee_{(i,j) \in P} (1 - \mu_{ij}) = \bigwedge_{(i,j) \in P} \mu_{ij}$ é o valor de pertinência do caminho P e $c(P)$ seu custo. Este procedimento não reduz o número e iterações do algoritmo proposto na seção 5.4.1, mas o caminho apresentado como solução por este modelo deve considerar comprimento e pertinência associados, de modo que o balanço seja o *melhor* possível.

O número de iterações seria o mesmo que o algoritmo proposto, já que é necessário conhecer o conjunto de caminhos possíveis e sua pertinência. A idéia é escolher o caminho que possui menor comprimento e maior confiabilidade.

5.4.5 Encontrar o caminho com maior valor de pertinência: CM-Max

Ainda para o caso de um grafo denso de grandes dimensões em que o conjunto de soluções fuzzy pode ser de pouca utilidade, pode-se tentar obter apenas o caminho mais confiável, ou seja, o caminho de maior pertinência ao conjunto fuzzy de soluções. Considerando os conceitos de grafos fuzzy, este é um algoritmo para encontrar o caminho mais forte entre dois nós (Capítulo 3).

Neste sentido, o algoritmo de Bellman-Ford-Moore (Goldbarg e Luna, 2000) modificado será utilizado para encontrar a solução. Ao invés de usar o comprimento do caminho para calcular o comprimento mínimo, usaremos para encontrar o maior valor de pertinência, μ_{ij} , ou o menor complemento da pertinência: $(1 - \mu_{ij})$.

Seja $m_j^k = [i, \mu_{sj}]$ a etiqueta associada ao nó j , onde i é o nó predecessor e μ_{sj} é a pertinência encontrada para o caminho entre o nó origem s e o nó j . k designa a iteração.

Passo 0: Seja um grafo $G : (N, A)$, μ_{ij} o valor de pertinência do arco (i, j) . Faça $m_s^0 \leftarrow [0, 1]$ e $m_j^0 \leftarrow [-, 1]$, $j \in N$, $j \neq s$. Faça $k \leftarrow 1$.

Passo 1: Faça:

- $m_j^{k+1} \leftarrow \min\{m_j^k, \max_i\{\min\{m_i^k, \mu_{ij}\}\}\}$, $i \neq j, s$, $j \in N$.
- $k \leftarrow k + 1$.

Se $m_j^{k-1} \equiv m_j^k$, $\forall j \in N$, então FIM. Caso contrário, repita o Passo 1.

Dado um nó i , o algoritmo atualiza os valores das etiquetas referentes ao nó predecessor que faz parte do caminho com maior grau de pertinência (ou caminho mais forte) e o grau de pertinência deste caminho. Quando não houver mais atualizações, o procedimento é encerrado e o valor do caminho de maior confiabilidade de um caminho do nó origem s até o nó destino t e a todos os outros nós, é devolvido ao decisor.

5.5 Exemplos numéricos

Seja o exemplo encontrado no artigo de (Okada e Soper, 2000) em que considera-se o mapa rodoviário dos Estados Unidos. O problema é encontrar o menor caminho entre Boston e Los Angeles (mapa do exemplo no Apêndice C). O digrafo associado G possui 70 nós e 210 arcos. Os algoritmos foram implementados no Matlab v6.1. Os valores de pertinência foram gerados de modo aleatório, com distribuição uniforme, para cada teste realizado.

Teste	rotas	$\bar{\mu}$	DVV90CM	CM-Max
1	09	0,6217	0,1200	0,0300
2	05	0,5579	0,1200	0,0200
3	02	0,3578	0,0900	0,0200
4	06	0,6337	0,1200	0,0500
5	11	0,6320	0,1800	0,0300

Tabela 5.3: Testes realizados para o exemplo de Okada e Soper, 2000

Na Tabela 5.3 temos a quantidade de rotas encontrada pelo algoritmo DVV90CM, o maior valor de pertinência encontrado dentre as rotas, os tempos de execução (em segundos) do algoritmo exato (DVV90CM) e do algoritmo de maior pertinência (CM-Max). Como apenas os valores de pertinência do arco variaram nos testes da tabela 5.3, nota-se a influência destes valores no número de rotas encontradas entre o nó 1 (Boston) e o nó 70 (Los Angeles).

Os resultados das heurísticas, para o problema da Tabela C.2, estão nas tabelas 5.4, 5.5 e 5.6

Teste	$\bar{\mu}$	T_{H1}
01	0,5645	0,0400
02	0,5509	0,0600
03*	0,3578	0,0400
04	0,6377	0,0700
05	0,5297	0,0800

Tabela 5.4: Heurística 1: Resolver para um valor de $\alpha = 0,5$. Teste 3: $\alpha = 0,3$

Na heurística 3, tabela 5.6, escolhe-se a melhor relação (confiabilidade) \times (comprimento). Para os testes 03, 04 e 05, foi escolhido o caminho com maior pertinência encontrada. Para os teste 01 e 02 foram escolhidos caminhos intermediários, isto é, estes caminhos devem ter um comprimento menor o suficiente para compensar a confiabilidade, segundo o critério adotado.

Teste	rotas	$\bar{\mu}$	T_{H_2}
01	02	0,4159	0,0200
02	02	0,4663	0,0300
03	02	0,3578	0,0200
04	02	0,4701	0,0300
05	02	0,4326	0,0200

Tabela 5.5: Heurística 2: Determinar o número máximo de partições: $p = 3$

Teste	$\bar{\mu}$	T_{H_3}
01	0,5888	0,1300
02	0,4663	0,1250
03	0,3578	0,0900
04	0,6337	0,1200
05	0,6320	0,1900

Tabela 5.6: Heurística 3: Solução com melhor relação (valor de pertinência) \times (comprimento do caminho)

Para observar o comportamento das heurísticas e dos algoritmos propostos, foram gerados dois grafos, G500 e G1000, $G = (N, A)$, tal que $(i, j) \in A \rightarrow i < j, i, j \in N$, custo dos arcos $c_{ij} \in [0, 100]$ e valor de pertinência aleatório $\mu_{ij} \in [0, 1]$. O grafo G500 possui 500 nós e 34.032 arcos. O grafo G1000 possui 1000 nós e 135.597 arcos. Os testes 01, 02 e 03 da tabela 5.7 são referentes ao grafo G500 e os testes 04, 05 e 06 referentes ao grafo G1000, para diferentes valores

Teste	rotas	$\bar{\mu}$	T_e	T_m
01	14	0,9781	43,2520	12,5980
02	09	0,9698	33,2780	18,1760
03	17	0,9720	50,2120	11,6560
04	09	0,9885	181,7720	75,3890
05	12	0,9672	174,1500	89,2790
06	14	0,9879	461,3240	93,3440

Tabela 5.7: Testes com os grafos G500 e G1000

de pertinência. São dados o número de rotas, o maior grau de pertinência obtido e o tempo de execução (em segundos) dos algoritmos DVV90CM e CM-Max.

Pode-se notar a influência do tamanho dos grafos e dos parâmetros associados. O algoritmo

que calcula o caminho com maior confiabilidade, CM-Max, necessita de menos da metade do tempo para encontrar a solução, mas o algoritmo DVV90CM encontra todo o conjunto solução. Cabe ao decisor escolher o algoritmo que melhor atende suas necessidades.

Escolhendo os testes realizados com o grafo G1000 para testar as heurísticas, temos as tabelas 5.8, 5.9 e 5.10:

Teste	$\bar{\mu}$	T_{H1}
04	0,9272	3,1840
05	0,9054	2,7040
06	0,9010	2,6940

Tabela 5.8: Heurística 1: Resolver para um valor de confiabilidade escolhido: $\alpha = 0,9$

Teste	rotas	$\bar{\mu}$	T_{H2}
04	04	0,9272	140,6720
05	03	0,8615	115,1860
06	05	0,9010	213,4970

Tabela 5.9: Heurística 2: Determinar o número máximo de partições: $p = 5$

As heurísticas apresentadas na Tabela 5.8 e 5.9 não encontraram o caminho de maior confiabilidade, mas obtiveram seus resultados em tempo reduzido comparado aos algoritmos propostos anteriormente. Observando a Tabela 5.8 verificamos que o tempo é consideravelmente inferior aos outros algoritmos testado. É realizado apenas um α -corte e o caminho é encontrado através do algoritmo de Dijkstra. Mas o resultado é condicionado à escolha deste α -corte, o que pode ser difícil dependendo do problema. Já pelos dados da heurística 2, o tempo é reduzido, mas o tempo computacional é dependente da quantidade máxima de caminhos que se deseja obter.

Teste	$\bar{\mu}$
04	0,9815
05	0,6161
06	0,9879

Tabela 5.10: Heurística 3: Solução com melhor relação (valor de pertinência) \times (comprimento do caminho)

Para a heurística 3, encontramos a segunda maior pertinência para o teste 04, o menor valor de pertinência dentre os caminhos do teste 05 e o maior valor de pertinência para o teste 06, o

que nos faz observar que o caminho a ser escolhido pela heurística depende da variação do custo e da relação observada entre valor de pertinência e comprimento do caminho.

Como cada algoritmo apresentado neste trabalho possui características próprias e atende a uma determinada necessidade de solução, cabe ao decisor observar qual deles é mais apropriado para seu uso.

Capítulo 6

Árvore Geradora Mínima Fuzzy

Neste capítulo, a incerteza será considerada em dois casos: no primeiro caso a estrutura do grafo é crisp e os parâmetros são números fuzzy e no segundo caso temos que encontrar uma árvore geradora mínima em um grafo fuzzy. A resolução do primeiro caso será baseada na teoria de possibilidade (Zadeh, 1978), de forma a encontrar todo o conjunto solução fuzzy do problema. Serão propostos um algoritmo para a solução ótima e um algoritmo genético para tentar contornar a questão da complexidade do problema. Para o segundo caso, os algoritmos encontrados na literatura serão estudados e, com base nas observações feitas, melhorias serão propostas.

6.1 Introdução

O problema da árvore geradora mínima aparece em uma série de aplicações, ou como um subproblema destas. Um exemplo é a instalação de linhas telefônicas (ou elétricas) entre um conjunto de localidades utilizando a infra-estrutura das rodovias com o menor uso de material. Outros problemas (análise de clusters, armazenamento de informações, dentre outros) também podem ser tratados por esta modelagem, que possui eficientes algoritmos como Kruskal, Prim e Sollin (Ahuja et al., 1993).

Nos problemas reais, parâmetros como tempo e capacidade associados podem ter naturezas incertas. Portanto, estes problemas podem ser estudados e resolvidos sob a ótica de conjuntos fuzzy (Zadeh, 1965). Se considerarmos a segurança de uma rede de telecomunicações, as incertezas também podem estar associadas a estrutura. Também neste caso, a teoria de conjuntos

fuzzy pode ser associada como foi feito inicialmente por (Rosenfeld, 1975).

Na Seção 6.1.1 faremos uma definição do caso clássico e de suas condições de otimalidade e na Seção 6.1.2 serão apresentados os estudos encontrados na literatura sobre a incerteza associada a problemas da árvore geradora mínima. Na Seção 6.2 é estabelecido o problema da árvore geradora mínima com parâmetros fuzzy. Nas seções 6.2.3 e 6.2.4 serão apresentados os conceitos necessários para a construção de um algoritmo *exato* e um algoritmo genético para o problema da árvore geradora mínima com parâmetros fuzzy. Para o problema da árvore geradora mínima em um grafo fuzzy, a Seção 6.3 estuda os algoritmos encontrados na literatura e propõe outros algoritmos e heurísticas.

6.1.1 Definição do caso clássico

Seja $G = (N, A)$ um grafo conectado com m nós e n arestas com custo c_{ij} associado à cada aresta ($c_{ij} \geq 0$). O problema consiste em construir uma árvore geradora de menor custo entre todos os nós (vide Figura 6.1).

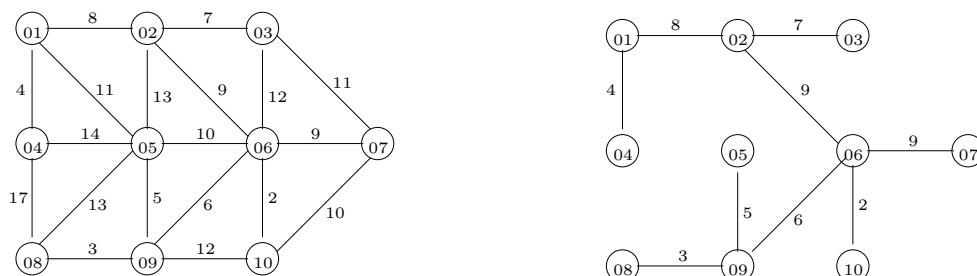


Figura 6.1: Exemplo de um problema da árvore geradora mínima

Uma árvore geradora T^* é uma solução ótima para o problema se, e somente se, satisfaz as seguintes condições:

Corte: Para toda aresta da árvore $(i, j) \in T^*$, temos que $c_{ij} \leq c_{kl}$ para toda aresta (k, l) contido no corte formado pela retirada de um aresta de (i, j) de T^* .

Caminho: Para toda aresta $(k, l) \in G$, $(k, l) \notin T^*$, temos que $c_{ij} \leq c_{kl}$ para cada aresta (i, j) contido no caminho em T^* que conecta o nó k ao nó l .

Para o caso clássico, o problema da árvore geradora mínima possui eficientes algoritmos de resolução, como por exemplo o algoritmo Kruskal, de Prim e o algoritmo de Sollin. O Capítulo

13 de (Ahuja et al., 1993) possui um estudo sobre este problema.

6.1.2 Estudos encontrados na literatura

Foram encontrados dois trabalhos tratando do problema da árvore geradora mínima fuzzy. Cada um estuda um caso diferente: o de grafo crisp com parâmetros fuzzy e o de grafo fuzzy com parâmetros crisp.

Em (Chunde, 1996), a estrutura considerada tem nós crisp (isto é $\sigma(i) = 1, \forall i \in N$) e arestas com graus de pertinência associados (isto é, $\mu_{ij} \in [0, 1], \forall (i, j) \in A$). São propostos três algoritmos considerando algumas situações da rede, sem considerar o custo das arestas como números fuzzy.

(Chang e Lee, 1999) analisaram os problemas da árvore geradora mínima, PERT e de caminho mínimo, utilizando algoritmos clássicos e um método proposto pelos autores (OERI), para o ordenamento de números fuzzy em problemas de redes. Este resolve o problema utilizando um algoritmo crisp para obter a solução.

Nas próximas seções novos algoritmos serão propostos com base nos estudos destes trabalhos e do Capítulo 4.

6.2 Caso I: problema com grafo crisp e parâmetros fuzzy

No problema da árvore geradora mínima com o grafo crisp e parâmetros fuzzy, temos que o custo (ou comprimento) associado a cada aresta é um número fuzzy. Veja o exemplo da Figura 6.2 em que o custo associado a cada aresta é um número triangular fuzzy $(\underline{c}, c, \bar{c})$:

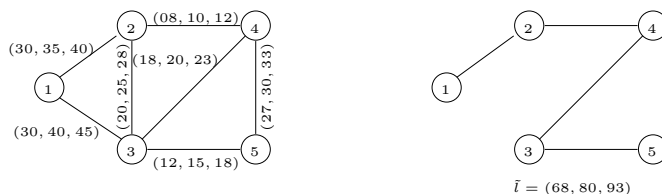


Figura 6.2: Exemplo de um problema da árvore geradora mínima com parâmetros fuzzy

A solução T^* encontrada pelo algoritmo proposto por (Chang e Lee, 1999) (também na Figura 6.2) fornece a mesma árvore para os três casos considerados no trabalho citado: indiferente,

otimista e pessimista. Basicamente, o caso otimista é baseado no valor \underline{c} , o caso pessimista é baseado no valor \bar{c} e o caso indiferente no valor modal c . O método OERI proposto no trabalho propõe um cálculo, ponderado, semelhante ao do centro de massa.

Agora, vamos considerar as árvores geradoras da Figura 6.3:

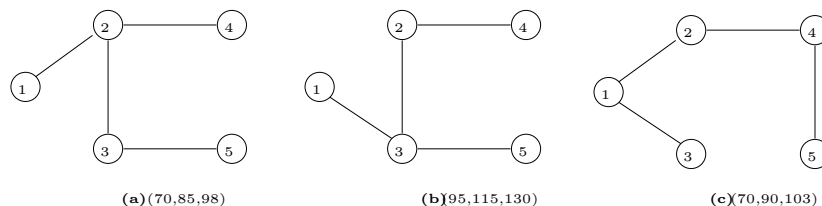


Figura 6.3: árvores geradoras

sendo que (a) tem comprimento (70, 85, 98), (b) (70, 90, 103) e (c) (95, 115, 130).

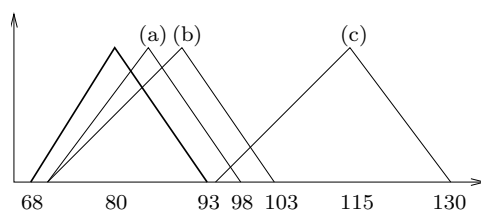


Figura 6.4: Comparação dos comprimentos fuzzy

Com base na observação da Figura 6.4, percebemos que não existe possibilidade da árvore geradora (c) ser custo menor do que a solução encontrada previamente na Figura 6.2. Já para as árvores (a) e (b) existe uma possibilidade para que cada uma destas árvores seja menor do que a solução encontrada. Podemos dizer então que existem vários *graus de possibilidade* de uma árvore pertencer ao conjunto solução do problema. (Blue et al., 2002) e (Chanas, 1987) entendem este valor como o grau de pertinência desta solução no conjunto fuzzy de soluções.

Nossa proposta é trabalhar com a teoria de possibilidade (Zadeh, 1978) para tratar a solução deste problema. Dois pontos de vista serão considerados:

1. Encontrar a árvore geradora mínima fuzzy de maior grau de possibilidade (extensão do caso crisp).
2. Encontrar uma solução parcial, isto é, um conjunto de árvores, cada uma com um grau de possibilidade de ser a árvore geradora mínima (fuzzy) em relação à solução de grau máximo.

6.2.1 Parte 1: Encontrar a árvore geradora mínima com custo fuzzy de maior grau de possibilidade

Para obter a árvore geradora com maior grau de possibilidade de ser a árvore geradora mínima devemos observar as propriedades referentes a comparação de números fuzzy. Utilizando a Proposição 2.7.1 no Capítulo 2, podemos determinar qual a árvore geradora mínima com maior grau de possibilidade. Esta proposição diz que pela comparação dos valores modais de dois números fuzzy podemos verificar qual deles tem o valor de possibilidade igual a 1. Então o número fuzzy com menor valor modal deve ser associado a solução com maior grau de possibilidade. Com isso, resolvendo o problema crisp associado (grafo G e custo igual ao valor modal do custo fuzzy), encontraremos a árvore geradora de maior valor de possibilidade de ser a árvore geradora mínima.

O teorema 2.7.1 pode ser interpretado como uma extensão da abordagem crisp do problema da árvore geradora mínima e resolve o primeiro caso a ser estudado.

6.2.2 Parte 2: Encontrar o conjunto fuzzy de soluções do problema da árvore geradora mínima com custo fuzzy

Para a construção da árvore geradora fuzzy devemos obter todas as árvores geradoras com algum grau de possibilidade de ser a árvore geradora mínima. Enumerar árvores geradoras torna o problema não polinomial (Seção 4.1.2). De fato, o número de árvores geradoras em um grafo completo com m nós é da ordem de m^{m-2} (Julstrom e G.R.Raidl, 2002). Portanto, deve desenvolver técnicas que promovam um corte no espaço de soluções e evitem cálculos desnecessários durante este processo.

Para tentar resolver estas questões, será utilizada a árvore geradora mínima fuzzy de maior grau de possibilidade (parte 1, Seção 6.2.1) para gerar as outras árvores. Com isso deve-se privilegiar a construção das árvores geradoras com grau de possibilidade não nulo, isto é, que possuem alguma chance de serem a árvore geradora mínima, pois as árvores que possuem partes em comum costumam ter maiores graus de possibilidade (Okada e Soper, 2000).

Seja o exemplo da Figura 6.2. O grafo G apresentado tem como solução a árvore geradora T^* . Usando o critério de otimalidade baseado no conjunto de corte $[S, \bar{S}]$, podemos obter outras árvores geradoras. Por exemplo, retirando a aresta $(3, 4)$ temos três arestas pertencentes ao conjunto de corte (Figura 6.5):

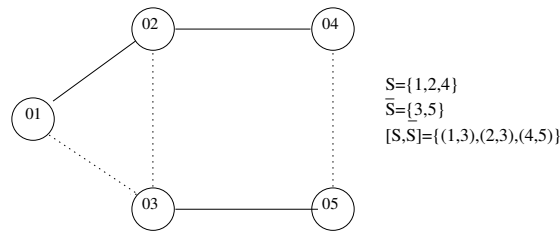


Figura 6.5: Arcos que podem substituir a aresta (3,4)

Cada aresta que pode substituir a aresta (3,4) cria uma nova árvore geradora. Cada uma das três possíveis árvores geradoras estão representadas na Figura 6.6, com seus respectivos custos fuzzy:

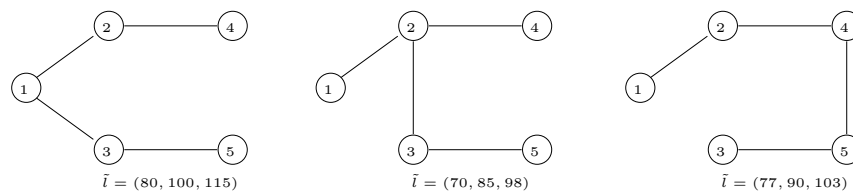


Figura 6.6: Árvores criadas pela substituição da aresta (3,4) na árvore T^*

Portanto, se retirarmos k arestas ($k = 1, \dots, n - 1$) da árvore T^* , podemos encontrar outras árvores geradoras. Quando $k = n - 1$, todas as arestas de T^* serão retiradas. Então resolvemos o problema com o novo grafo $G' = G - T^*$ e continuamos a enumeração a partir de T' , comparando as novas árvores com T^* até a enumeração completa das árvores, quando $|A| < |N| - 1$.

Podemos calcular a possibilidade de cada uma destas árvores serem menores do que T^* durante o processo de construção das árvores geradoras. Por exemplo, quando retiramos a aresta (3,4) na Figura 6.5, encontramos três arestas que vão construir outras três árvores. Quando for feito o cálculo da possibilidade de uma destas ser menor que T^* , o cálculo será feito apenas entre a aresta (3,4) e cada aresta do corte, segundo definição de possibilidade dada por (Okada, 2001). Portanto, o cálculo de possibilidade será realizado entre os k arestas que diferenciam a árvore geradora T^* e as demais árvores.

A construção das árvores pode ser orientada de forma que o descarte das árvores que não farão parte do conjunto solução (possibilidade nula) sejam facilitadas. Usando o teste de corte proposto em (Blue et al., 2002) para o caso de caminho mínimo, podemos descartar árvores

geradoras com as seguintes características:

$$\sup\{\text{supp}\{\tilde{l}_{\bar{T}}\}\} < \inf\{\text{supp}\{\tilde{l}_{T_k}\}\}, \quad \forall T_k \in \mathcal{T}$$

onde \bar{T} é a árvore encontrada a partir do grafo crisp \bar{G} ($\bar{c}_{ij} = \sup\{\text{supp}\{\tilde{c}_{ij}\}\}$, $(i, j) \in A$) e \mathcal{T} é o conjunto de árvores geradoras de G . Esta característica está demonstrada na Proposição 2.7.2.

Uma sugestão para que o descarte seja feito de modo mais eficiente, no momento em que o número de arestas for igual a $k = |N| - 1$, a próxima árvore geradora a ser a base para a enumeração de outras será calculada a partir de \underline{G} ($\underline{c}_{ij} = \inf\{\text{supp}\{\tilde{c}_{ij}\}\}$). A árvore geradora fuzzy \underline{T} composta pelas mesmas arestas da solução crisp de \underline{G} será comparada a T^* . Caso esta árvore seja descartada pelo critério de corte dado acima, então todas as árvores geradoras que ainda não foram enumeradas também serão descartadas.

As idéias apresentadas nas seções 6.2.1 e 6.2.2 são a base para o algoritmo que será proposto na próxima seção.

6.2.3 Algoritmo AGMCF:

Este algoritmo está separado em três procedimentos. O primeiro é o programa principal:

Procedimento 1:

Passo 0: (*Iniciar os parâmetros*) Encontre a árvore geradora mínima fuzzy T^* associada a solução do grafo crisp G_C com custo crisp $c_{ij} = \inf_{\mu_{ij}(u)=1}\{u\}$. Faça $\delta = \sup\{\text{supp}\{\tilde{l}_{T^*}\}\}$.

Passo 1: (*Encontrar árvores com arestas coincidentes a T^**) Para $k = 1$ até $k = |N| - 2$, faça:

- Para cada k arestas da árvore T^* (estes k arestas $\in R'$, conjunto das arestas retirados da árvore), execute o procedimento 2.

Passo 2: (*Encontrar árvores sem arestas coincidentes com T^**) Faça:

1. $A_1 \leftarrow A - T^*$. Seja $G_1 = (N, A_1)$.
2. Encontre T_1 através da resolução do grafo crisp \underline{G}_1 com custo $c_{ij} = \inf\{\text{supp}\{\tilde{c}\}\}$.
3. Se não houver mais árvores geradoras ou se $\inf\{\text{supp}\{\tilde{l}_{T_1}\}\} > \delta$, então FIM. Caso contrário, para $k = 1$ até $|N| - 2$, faça:
 - Para cada k arestas da árvore T_1 (estes k arestas $\in R'$), execute o procedimento 3.

Procedimento 2: (*Construir as árvores geradoras e calcular a possibilidade*)

Passo 0: (*Inicialização dos parâmetros*) Sejam k , T e R' dados. Faça $\tilde{l}' \leftarrow \sum_{(i,j) \in R'} \tilde{c}_{ij}$.

Passo 1: (*Comparação entre arestas*) Para cada k arestas $(s, t) \in [S, \bar{S}]$ (estas k arestas $\in R''$) que não formam ciclo com $(T - R')$, faça:

- $\tilde{l}'' \leftarrow \sum_{(i,j) \in R''} \tilde{c}_{ij}$
- Se $\inf\{\tilde{l}_T - \tilde{l}' + \tilde{l}''\} > \delta$, então descarte esta árvore. Caso contrário, $w \leftarrow Poss(l'' \leq l')$.
- Faça $D_{ij} \leftarrow \max\{D_{ij}, w\}$, $(i, j) \in R''$.

Passo 2: Caso não existam mais árvores geradoras a serem formadas, volte ao Passo 1 do procedimento 1.

A diferença entre o segundo e o terceiro procedimentos é o cálculo da possibilidade (Equação 2.6). O segundo procedimento considera as árvores interativas, e usa o cálculo da possibilidade definido por (Okada, 2001). O terceiro procedimento usa o cálculo clássico de (Dubois e Prade, 1980).

Procedimento 3: (*Construir as árvores geradoras e calcular a possibilidade*)

Passo 0: (*Inicialização dos parâmetros*) Sejam k , T , \tilde{l}_{T^*} e R' dados. Calcule \tilde{l}_T .

Passo 1: (*Comparação com a árvore geradora mínima fuzzy T^**) Para cada k arestas $(s, t) \in [S, \bar{S}]$ (estes k arestas $\in R''$) que não formam ciclo com $(T - R')$, faça:

- Calcule $\tilde{l}'' \leftarrow \tilde{l}_T - \sum_{(i,j) \in R'} \tilde{c}_{ij} + \sum_{(i,j) \in R''} \tilde{c}_{ij}$
- Se $\inf\{\tilde{l}''\} > \delta$, então descarte esta árvore. Caso contrário, $w \leftarrow Poss(l'' \leq l_{T^*})$.
- Faça $D_{ij} \leftarrow \max\{D_{ij}, w\}$, $(i, j) \in T$.

Passo 2: Caso não existam mais árvores geradoras a serem formadas, volte ao passo 2 procedimento 1.

Mesmo com os testes de corte previsto no algoritmo, a enumeração das árvores que farão parte da solução ainda torna o problema de difícil resolução. No algoritmo temos que escolher os subconjuntos de arestas a serem retirados e quais grupos podem substituir cada um destes subconjuntos. Isto faz com que o problema ainda seja combinatorial.

De fato, o número de árvores geradoras é da ordem de m^{m-2} , $m = |N|$ (Raidl e Julstrom, 2001) em um grafo completo. No exemplo da Figura 6.2, com 05 nós e 07 arestas, existem 21

árvores geradoras. Em um grafo completo com 05 nós temos 125 árvores. Se for observado o fato de que a solução é composta por árvores com maior grau de possibilidade, pode-se propor uma heurística para encontrar uma solução aproximada satisfatória para o problema. Neste contexto, um algoritmo genético será construído com base nas características do problema e da solução.

6.2.4 Um algoritmo genético para o problema da árvore geradora mínima com parâmetros fuzzy

Com base nos estudos realizados nas Seções 6.2.1 e 6.2.2, temos que o problema da árvore geradora mínima com parâmetros fuzzy possui as seguintes características:

1. Um conjunto fuzzy de soluções em que cada árvore geradora possui uma *possibilidade* de ser a árvore geradora mínima.
2. Necessidade de avaliar uma grande quantidade de árvores geradoras para construção do conjunto fuzzy de soluções.
3. As árvores geradoras pertencentes ao conjunto solução normalmente tem arestas em comum (Okada e Soper, 2000).

Estas características levaram a proposta de desenvolver um algoritmo genético (Apêndice B) para tentar encontrar uma boa solução para o problema proposto. A população é formada por indivíduos (para este problema, as árvores geradoras) que evoluem por meio de operadores de reprodução, que geram novos indivíduos (característica 2) que herdam determinadas características dos indivíduos-pais (característica 3). Procurando um subconjunto fuzzy de soluções, a medida de adaptação (também conhecido como função de fitness) é utilizada de forma a privilegiar a reprodução dos indivíduos com maior valor de fitness (no caso, o valor de possibilidade da árvore geradora ser a árvore geradora mínima).

Este algoritmo foi baseado nos estudos (Raidl e Julstrom, 2001) e (Julstrom e G.R.Raidl, 2002) para o problema da árvore geradora mínima com restrições de capacidade.

Codificação: Cada cromossomo representa uma árvore geradora. Seja um grafo $G : (N, A)$ e seja $m = |N|$. Uma árvore geradora é representada por $m - 1$ arestas, que conectam todos os nós e não formam ciclos. Um cromossomo representando uma árvore geradora possui então $m - 1$ genes, cada gene contém a informação de qual aresta pertence a árvore geradora (Figura 6.7, as arestas estão numeradas para facilitar a visualização).

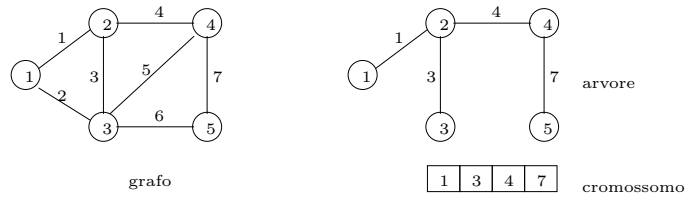


Figura 6.7: Exemplo de um cromossomo representando uma árvore

Inicialização: As arestas não podem ser selecionadas de forma aleatória pois isto pode resultar em um subgrafo desconexo. Este é um caso em que o genótipo (cromossomo) é influenciado pelas características desejadas (fenótipo) ao indivíduo. Portanto, a inicialização é feita com uma adaptação do algoritmo de Kruskal: o algoritmo adaptado não ordena as arestas pelo custo, pelo contrário, escolhe as arestas de forma aleatória para formar a árvore geradora.

Recombinação: Esta operação escolhe dois indivíduos da população, T_1 e T_2 , como os pais e constrói um subgrafo $T = (N, T_1 \cup T_2)$. Uma nova árvore é obtida deste subgrafo T a partir do algoritmo adaptado de Kruskal utilizado na inicialização (Figura 6.8).

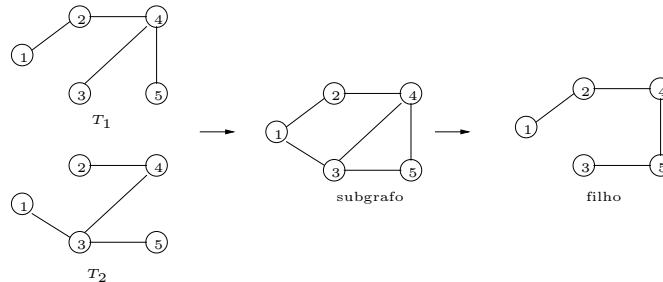


Figura 6.8: Recombinação de duas árvores

Mutação: após a escolha de um indivíduo (árvore geradora), um determinado número de bits (arestas) do indivíduo são proibidos de pertencer ao novo indivíduo. Outras arestas são escolhidas para substituí-las, sem que haja a formação de ciclos. (Figura 6.9).

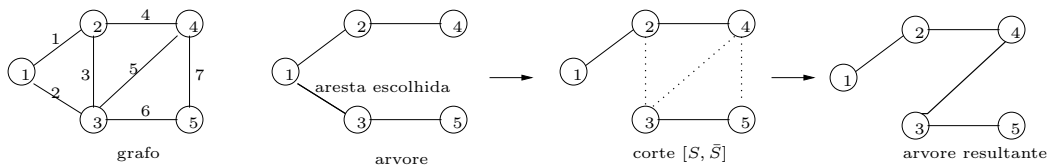


Figura 6.9: Mutação em uma árvore

Outras informações:

- O valor do *fitness* (medida de adaptação do indivíduo ao meio) de cada cromossomo é o grau de possibilidade deste ser menor que a árvore geradora mínima de maior grau de possibilidade (ver Seção 6.2.1).
- A escolha dos indivíduos na recombinação e na mutação é feita em parte pela roleta e em parte de forma aleatória.
- A nova população é formada pelas árvores geradoras formadas pela recombinação (40%), por mutação (20%) e por indivíduos escolhidos da população anterior, via roleta e de forma aleatória.
- A diversidade da população é importante para tentar fazer uma cobertura ampla do nosso espaço de busca. Por isso a taxa de mutação tem um valor maior que o utilizado em algoritmos genéticos convencionais.

A atualização do grau de possibilidade de cada aresta durante as gerações da população é realizada cada vez que um novo indivíduo é criado, pela seguinte fórmula:

$$D_{ij} = \max_{ij \in u} \{D_{ij}, fitness(u)\} \quad (6.1)$$

sendo u um novo indivíduo que contem a aresta (i, j) . No final das gerações, D fornecerá o grau de possibilidade de cada aresta pertencer à árvore geradora mínima.

A estrutura do algoritmo genético é dado na seção abaixo:

6.2.5 Algoritmo genético AGMCF-AG:

Passo 0: Criar a população inicial, com K indivíduos, pelo método de Kruskal adaptado. Seja P a população. Calcule o valor do *fitness* de cada indivíduo.

Passo 1: Para $I = 1$ até GER ou quando não houver mudanças em D em um determinado número de gerações, faça:

1. Selecione uma percentagem dos indivíduos de P para recombinação, proceda a operação e aloque os novos indivíduos na população intermediária (P_i).
2. Selecione uma percentagem dos indivíduos de P para a mutação, proceda a operação e aloque os indivíduos na P_i .
3. Selecione o melhor indivíduo da P e sorteie os restantes dos indivíduos que devem completar a P_i .

4. Faça $P \leftarrow P_i$. Calcule o valor do fitness de cada indivíduo de P .
5. Atualize D_{ij} (Equação 6.1).

Passo 2: Devolva D ao decisor.

6.2.6 Testes e análises

Utilizando o exemplo dado anteriormente (Figura 6.2), temos a solução de grau máximo com comprimento (68, 80, 93). O algoritmo proposto na Seção 6.2.3, AGMCF, analisou as 15 das 20 árvores geradoras (por exemplo, as da Figura 6.3), excluindo a árvore de máximo grau de possibilidade. As cinco árvores restantes não foram consideradas por terem os valores de possibilidade nulos. Por ser um exemplo pequeno, com poucos indivíduos, o algoritmo genético AGMCF-AG encontrou a solução em todos os testes realizados com população de 10 indivíduos em 50 gerações.

A solução do problema é dada na Figura 6.10. Os valores apresentados sobre cada aresta são os graus de possibilidade das arestas:

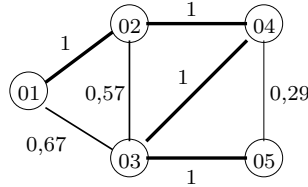


Figura 6.10: Solução do problema

Em (Ali et al., 2000) encontramos uma rede óptica de telecomunicações com as respectivas distâncias entre as cidades. Considerando os custos fuzzy como números triangulares do tipo LR com valor modal igual ao comprimento das arestas e desvio de 10% em relação a este valor (Tabela C.1, no Apêndice C), chega-se à seguinte solução (Tabela 6.1) para o algoritmo AGMCF e do genético AGMCF-AG.

Ambos os algoritmos foram implementados em MATLAB 6.1. O tempo gasto com o AGMCF-AG foi insignificante (aproximadamente 70 segundos) em relação ao algoritmo AGMCF (em dias) e a solução do algoritmo genético está a 97,72% em média do resultado ótimo (variando entre 94,64% e 99,01% do ótimo, em 10 execuções do programa). Foram analisadas cerca de 4000 árvores pelo algoritmo genético em um conjunto de soluções que tem como limitante superior $2,2 \times 10^{11}$ candidatos, considerando as combinações de arestas a serem retirados e as combinações

Arestas	AGMCF	AGMCF-AG	Arestas	AGMCF	AGMCF-AG	Arestas	AGMCF	AGMCF-AG
1	0,5251	0,5192	13	0,4615	0,4586	25	1,0000	1,0000
2	1,0000	1,0000	14	0,7674	0,7674	26	0,6753	0,6753
3	0,2188	0,1704	15	1,0000	1,0000	27	0,2714	0,2714
4	1,0000	1,0000	16	0,5550	0,5395	28	0,1731	0,0664
5	0,8718	0,8718	17	1,0000	1,0000	29	1,0000	1,0000
6	1,0000	1,0000	18	1,0000	1,0000	30	0,3056	0,2829
7	1,0000	1,0000	19	0,3919	0,3750	31	0,1404	0,1084
8	1,0000	1,0000	20	0,3498	0,3243	32	1,0000	1,0000
9	1,0000	1,0000	21	1,0000	1,0000	33	1,0000	1,0000
10	1,0000	1,0000	22	1,0000	1,0000	34	1,0000	1,0000
11	1,0000	1,0000	23	1,0000	1,0000	35	1,0000	1,0000
12	0,6032	0,6032	24	0,6565	0,6565	36	1,0000	1,0000

Tabela 6.1: Graus de possibilidade obtidos pelos algoritmos ótimo e genético para a rede Itália

das arestas candidatos a substituí-los. Em um grafo completo com o mesmo número de nós, teríamos cerca de $1,3 \times 10^{25}$ possíveis árvores geradoras.

Contudo, o resultado pode ser melhorado. Pode-se implementar técnicas que garantam a diversidade da população e façam uma exploração mais eficiente do espaço de busca no caso do algoritmo genético.

Algumas características dos resultados do algoritmo AGMCF, como a necessidade de uma maior troca de arestas para se obter uma melhoria, ao invés de apenas uma troca considerada na mutação clássica do algoritmo genético, nos leva a considerar técnicas mais recentes de computação natural, como as técnicas de sistemas imunológicos artificiais (Silva, 2001) para tentar obter uma solução mais próxima a solução *ótima* do problema. Outra característica, a de evitar árvores com grau de possibilidade nula através dos cortes ou da busca em regiões próximas da árvore geradora de maior grau de possibilidade para a geração de novas árvores também podem ser levados em consideração para a geração da população inicial.

6.3 Caso II: O problema da árvore geradora mínima em um grafo fuzzy

Seja um grafo fuzzy $G(\sigma, \mu)$, sendo σ a função de pertinência associada a conjunto de nós N e μ a função de pertinência associada às arestas A . Como foi visto no Capítulo 3, temos que $\mu_{ij} \leq \sigma_i \wedge \sigma_j$, $(i, j) \in A$ e $i, j \in N$, ou seja, o valor de pertinência de um arco (ou aresta) ao

grafo não pode ser maior que o valor de pertinência dos nós que este arco (ou aresta) está associado (Rosenfeld, 1975). Os custos associados ao grafo fuzzy também podem ser números fuzzy, mas neste trabalho consideraremos o custo crisp.

Uma árvore geradora, T , obtida em um grafo fuzzy deve ter um custo $c_T = \sum_{(i,j) \in T} c_{ij}$ e um valor de pertinência associado igual ao menor valor de pertinência de uma aresta pertencente a T : $\mu_T = \min_{(i,j) \in T} \mu_{ij}$. Este valor de pertinência pode ser interpretado como um valor de confiabilidade da árvore geradora. Isto pode ser válido em uma situação em que o valor de pertinência de cada aresta diz respeito a segurança de uma conexão em uma rede óptica, por exemplo.

Com isso, quando resolvemos o problema da árvore geradora mínima em um grafo fuzzy, obtemos um conjunto fuzzy em que cada elemento do conjunto é uma árvore geradora mínima com um grau de pertinência ao conjunto. Um algoritmo fundamental para problemas em grafos fuzzy é proposto em (Delgado et al., 1990). Neste trabalho, adaptamos o algoritmo fundamental para o problema da árvore geradora mínima (Seção 6.3.1). Em (Chunde, 1996), temos outro estudo encontrado na literatura que aborda algumas heurísticas para tratar o conjunto fuzzy de soluções. As idéias principais deste artigo estão apresentadas na Seção 6.3.2. A seguir, com base nas observações feitas sobre estes algoritmos, apresentamos duas propostas nas seções 6.3.3 e 6.3.4. Alguns exemplos sobre estes algoritmos são apresentados na Seção 6.3.6.

6.3.1 A adaptação do algoritmo de Delgado et. al., 1990

A idéia básica do algoritmo de (Delgado et al., 1990) é a utilização dos α -cortes para obter cada árvore geradora mínima do conjunto fuzzy de soluções. O algoritmo fundamental de (Delgado et al., 1990), está na Seção 4.2.1. As noções sobre α -cortes e outros conceitos da teoria de conjuntos fuzzy estão no Capítulo 2. Neste trabalho, todos os nós tem valor de pertinência igual a 1.

Iniciamos com o conjunto suporte G^* ($G^* : (N, A^*) | \mu_{ij} > 0, \forall (i, j) \in A$) do grafo G . Sendo o conjunto suporte um conjunto crisp, a solução (T_0) do problema baseado em G^* é encontrada por meio de um método clássico. Com isso, temos:

$$\alpha_0 = \min_{(i,j) \in T_0} \mu_{ij}$$

$$C_0 = \sum_{(i,j) \in T_0} c_{ij}$$

e o grau de pertinência da árvore (α_0) é obtido observando o valor de pertinência das arestas que foram selecionadas para a árvore geradora mínima. O custo C_0 é o menor valor encontrado para o grafo G .

Para obter as outras árvores geradoras com diferentes valores de pertinência, devemos retirar os arcos com $\mu_{ij} \leq \alpha_0$, isto é, deve-se proceder um α -corte no conjunto de arestas A . Com isso, temos um novo subgrafo de G (G_1^*), que deve ser conectado. Resolvendo G_1^* , outra árvore geradora mínima T_1 e seu grau de pertinência α_1 serão encontrados. Observe que:

$$\begin{aligned}\alpha_0 &< \alpha_1 < \dots < \alpha_k < \alpha_{k+1} < \dots < \alpha_R \\ C_0 &\leq C_1 \leq \dots \leq C_k \leq C_{k+1} \leq \dots \leq C_R\end{aligned}$$

onde R é a última iteração com o grafo conectado. Quando o subgrafo resultante não é mais conectado, o procedimento termina e o conjunto de soluções é apresentado ao decisor. A seguir, a adaptação do algoritmo de (Delgado et al., 1990), baseado nestes conceitos, é apresentado.

Algoritmo DVV90AGM:

Passo 0: Seja um grafo fuzzy $G = (\sigma, \mu)$ com custo $c_{ij} > 0$ para $(i, j) \in A$. Seja $k = 0$; $\alpha_0 = 0$.

Passo 1: Encontre o conjunto obtido pelo α -corte. Resolva o problema crisp associado. Seja T a árvore geradora mínima encontrada. Faça:

$$\begin{aligned}T_k &\leftarrow T & (6.2) \\ C_k &\leftarrow \sum_{(i,j) \in T} c_{ij} \\ \alpha_k &\leftarrow \bigwedge_{(i,j) \in T} \mu_{ij} = \min_{(i,j) \in T} (\mu_{ij})\end{aligned}$$

onde α_k é o valor de pertinência da árvore T_k com custo C_k .

Passo 2: Retire as arestas com valor de pertinência $\leq \alpha_k$ do grafo. Faça $k \leftarrow k + 1$ Se o grafo apresentar componentes não conectados \rightarrow FIM. Caso contrário volte ao Passo 1.

Complexidade do algoritmo

Neste caso, também teremos a complexidade do algoritmo dependendo do algoritmo crisp proposto e do número de arestas do problema. Considerando o algoritmo de Kruskal, a ordem de complexidade do problema é de $O(m+n \log n)$ (Ahuja et al., 1993). No pior caso, o procedimento deverá ter $n - m + 1$ iterações, ou seja, o algoritmo deve retirar apenas um aresta por vez, até chegar a uma árvore geradora. Portanto, a complexidade deste algoritmo será da ordem de $O((n - m)(m + n \log n))$.

A maior dificuldade do algoritmo DVV90AGM é a possibilidade de que o conjunto fuzzy solução ter uma grande dimensão e, com isso, ter pouca utilidade ao decisor. Neste sentido o trabalho de (Chunde, 1996) propõe duas heurísticas para o problema da árvore geradora mínima.

6.3.2 O trabalho de Chunde, 1996

Para tentar contornar a dificuldade encontrada no algoritmo DVV90AGM (Delgado et al., 1990), Chunde (1996) propôs realizar apenas uma iteração do algoritmo DVV90AGM, em um α -corte mais conveniente (heurística CH96a). Neste caso, o decisor escolhe um determinado grau de confiabilidade e para este valor é obtida a árvore geradora mínima correspondente. Outra proposta encontrada em (Chunde, 1996), resolve o seguinte problema (heurística CH96b):

$$\begin{aligned}
 \text{Min} \quad & g(T) \\
 \text{s.a.} \quad & g(T) = F(T)c(T) \\
 & F(T) = \bigvee_{(i,j) \in T} (1 - \mu_{ij}) \\
 & c(T) = \sum_{(i,j) \in T} c(i, j) \\
 & T \in \tau(N)
 \end{aligned} \tag{6.3}$$

em uma tentativa de equacionar a confiabilidade da árvore, μ_{ij} , e o seu custo total, $c(T)$, através de $g(T)$. Como o interesse é encontrar uma solução com mínimo custo e máxima pertinência, o complementar da pertinência μ_T foi utilizada. $\tau(T)$ é o conjunto de árvores geradoras. Basicamente ele também calcula as mesmas árvores geradoras mínimas que a proposta na Seção 6.3.1, mas determina a que tiver a melhor relação entre o custo da solução encontrada e o seu valor de pertinência, segundo critério proposto pelo autor.

6.3.3 Proposta 01: encontrar a árvore geradora fuzzy com maior grau de pertinência

Como, em geral, o custo total da árvore geradora mínima aumenta com o seu valor de pertinência, propomos encontrar a solução com o maior valor de pertinência possível. Com isso, o decisor tem disponível a solução *mais confiável*. Por outro lado, tendo os dois valores de pertinência extremos, ditos α_0 e α_R , pode-se optar por uma partição conveniente do problema, sabendo da sua monotonicidade. O limitante inferior é encontrado tomando o grafo G^* onde $\mu_{ij} > 0$.

Seja $G : (\sigma, \mu)$ um grafo fuzzy. Sejam σ e μ os valores de pertinência dos nós e arestas do grafo, respectivamente. Consideramos $\mu_{ij} = \mu_{ji} > 0$. A proposta do algoritmo é encontrar as arestas de maior pertinência de cada nó do grafo (Passo 1) e tomar o menor dentre estes valores de cada nó como o valor para o corte (Passo 2). Caso o grafo resultante seja não conectado, procuramos a aresta de maior pertinência que una estas partes (Passo 3). Com este procedimento, também sabemos o valor de pertinência da solução antes de resolvê-lo. Caso o usuário queira apenas saber o valor máximo de pertinência, o último passo é dispensável.

Algoritmo proposto para obter a árvore geradora mínima com maior valor de pertinência (MAX- α):

Passo 1: Para cada $i \in N$ faça $\beta_i \leftarrow \max_{j:(i,j) \in A} \{\mu_{ij}\}$

Passo 2: Seja $\alpha = \min_{i \in N} \{\beta_i\}$. Construa um grafo G' sendo

$$N' \leftarrow N$$

$$A' \leftarrow \{(i, j) \in A \mid \mu_{ij} \geq \alpha\}$$

Passo 3: Enquanto existirem componentes não conectadas em G' , faça:

Encontre a aresta $(p, q) \in A$ de maior valor de pertinência, μ_{pq} , que ligue duas componentes não conectadas em G' . Então, faça

$$A' \leftarrow A' \cup \{(i, j) \mid \mu_{pq} \leq \mu_{ij} \leq \alpha\}$$

$$\alpha \leftarrow \min\{\alpha, \mu_{pq}\}$$

Passo 4: Resolva o problema da árvore geradora mínima para G' por meio de um algoritmo clássico (Prim ou Kruskal). Se o problema tiver custos fuzzy, utilize algoritmo adequado.

Complexidade do algoritmo

Este algoritmo teria a sua complexidade próxima a do algoritmo de Prim, da ordem de $O(m + n \log n)$ (Ahuja et al., 1993), dependendo de como são implementados os passos 1 e 4.

6.3.4 Proposta 2: Heurística para obter um subconjunto fuzzy de soluções

Uma proposta corrente para os problemas em grafos fuzzy é a determinação do número de partições, h , pelo decisor. Com isso os α -cortes seriam previamente conhecidos e o número de soluções no subconjunto fuzzy seria limitado a h . Uma variante que propomos é a inclusão de um teste que evita o cálculo desnecessário em um α -corte. Caso a árvore geradora obtida em uma iteração k tenha uma pertinência $\delta > \alpha_{k+1}$, então o valor de $\alpha_{k+1} \leftarrow \delta$. Também, o intervalo a ser dividido está entre $[0, \alpha_R]$ e não entre $[0, 1]$.

6.3.5 Heurística (FMST-H1)

Passo 0: Seja um grafo fuzzy $G = (\sigma, \mu)$ com custo $c_{ij} > 0$, $(i, j) \in A$. Encontre a solução de máxima pertinência (MAX- α). Seja h o número de partições dado pelo decisor. Seja $\alpha_i = \frac{i \cdot \alpha_R}{h}$, $i = 1, \dots, h$. Faça $k = 0$ e vá para o passo 2.

Passo 1: Enquanto $\alpha_k \neq \alpha_R$ execute os passos 2, 3 e 4.

Passo 2: Resolva o problema crisp associado. Seja T a árvore geradora mínima encontrada. Faça:

$$\begin{aligned} T_k &\leftarrow T \\ C_k &\leftarrow \sum_{(i,j) \in T} c_{ij} \\ \alpha_k &\leftarrow \bigwedge_{(i,j) \in T} \mu_{ij} = \min_{(i,j) \in T} (\mu_{ij}) \end{aligned} \tag{6.4}$$

onde α_k é o valor de pertinência da árvore T_k com custo C_k .

Passo 3: Se $\alpha_k > \alpha_{k+1}$, faça $\alpha_{k+1} \leftarrow \alpha_k$ e $k \leftarrow k + 2$. Caso contrário, faça $k \leftarrow k + 1$. Retire as arestas com valor de pertinência menor ou igual a α_k do grafo. Se o grafo apresentar componentes não conectados, então finalize o programa e devolva o conjunto de solução ao decisor. Caso contrário volte ao Passo 1.

6.3.6 Exemplos e análises

Seja o grafo apresentado na Figura 6.1 com os seguintes valores de pertinência para suas arestas:

aresta	μ_{ij}	aresta	μ_{ij}	aresta	μ_{ij}	aresta	μ_{ij}
(1,2)	0,80	(2,6)	0,60	(5,6)	0,60	(6,10)	0,50
(1,4)	0,20	(3,6)	0,70	(5,8)	0,40	(7,10)	1,00
(1,5)	0,60	(3,7)	0,90	(5,9)	0,90	(8,9)	0,60
(2,3)	0,50	(4,5)	0,75	(6,7)	1,00	(9,10)	0,80
(2,5)	0,90	(4,8)	1,00	(6,9)	0,90		

Tabela 6.2: Pertinência das arestas no grafo G

A árvore geradora apresentada junto ao grafo na Figura 6.1 é a que possui menor custo, $c_0 = 53$, mas também menor pertinência, $\alpha_0 = 0,20$. Retirando as arestas com valor de pertinência menor ou igual a α_0 , obtemos uma outra árvore $T_1 = \{(1,2), (2,3), (2,6), (4,5), (5,9), (6,7), (6,9), (6,10), (8,9)\}$ que difere de T_0 somente pela troca da aresta (1,4) por (4,5), mas faz com que o custo aumente para $c_1 = 63$ e o valor de pertinência de T_1 para $\alpha_1 = 0,50$. Se retirarmos as arestas com valor de pertinência menor ou igual a α_1 , obtemos $T_2 = \{(1,2), (2,6), (3,7), (4,5), (5,9), (6,7), (6,9), (7,10), (8,9)\}$, com custo $c_2 = 75$ e valor de pertinência $\alpha_2 = 0,60$. Para $k = 3$: $T_3 = \{(1,2), (2,5), (3,7), (4,5), (4,8), (5,9), (6,7), (6,9), (7,10)\}$ com $c_3 = 93$ e $\alpha_3 = 0,75$ e esta é a árvore geradora mínima de maior grau de pertinência, pois a retirada da aresta com pertinência igual a torna o grafo desconexo.

Este exemplo é pequeno para aplicar as heurísticas, mas podemos verificar o funcionamento do algoritmo para obter a árvore geradora mínima de maior valor de confiabilidade. Seja $\beta = \{0.8, 0.9, 0.9, 1.0, 0.9, 1.0, 1.0, 1.0, 0.9, 1.0\}$ sendo β_i o maior valor de pertinência entre as arestas incidentes no nó i . Então o valor de corte é igual a $\alpha = \min_i \{\beta_i\} = 0.8$. Montando um subgrafo com as arestas com valor de pertinência maior que o valor de corte, notamos que existem duas componentes no subgrafo, uma englobando os nós 4 e 8 e a outra com os nós restantes. Tomando as arestas que conectam estas duas componentes, temos que a aresta (4,5) com valor de pertinência $\mu_{4,5} = 0,75$ é a aresta com maior valor de pertinência. Acrescentando as arestas que possuem valor de pertinência entre 0,75 e 0,80 obtemos um subgrafo em que apenas a árvore geradora mínima com maior valor de pertinência está contida.

Considere agora um grafo completo com 10 nós. As Tabelas C.3 e C.4 no apêndice C apresenta os dados deste grafo sendo que a primeira é referente ao valor de pertinência das arestas e a segunda aos custos.

No caso, o custo foi gerado aleatoriamente no Matlab v6.1, com valores entre 5 e 10, Os valores de pertinência tiveram o mesmo procedimento, com valores entre 0 e 1, distribuídos uniformemente. Utilizando ao algoritmo DVV90AGM (Delgado et al, 1990) obtemos um conjunto fuzzy de soluções com 13 árvores que são apresentadas com seus respectivos custos e valores de

pertinência.

	Árvore geradora mínima	custo	pertinência
1	{(1, 7), (4, 7), (1, 3), (1, 6), (5, 6), (5, 8), (2, 4), (3, 10)(4, 9)}	50	0,0579
2	{(1, 7), (4, 7), (1, 6), (5, 6), (5, 8), (2, 4), (4, 9), (5, 10), (3, 10)}	50	0,1509
3	{(1, 7), (4, 7), (1, 6), (6, 8), (5, 8), (2, 4), (4, 9), (5, 10), (3, 10)}	50	0,1934
4	{(1, 7), (4, 7), (2, 4), (4, 8), (5, 8), (6, 8), (4, 9), (5, 10), (3, 10)}	50	0,2987
5	{(1, 7), (4, 7), (2, 4), (4, 8), (5, 8), (6, 8), (4, 9), (2, 3), (3, 10)}	51	0,3529
6	{(1, 7), (4, 7), (2, 4), (4, 8), (5, 8), (6, 8), (4, 9), (3, 8), (3, 10)}	52	0,3704
7	{(1, 7), (4, 7), (2, 4), (4, 8), (6, 8), (4, 9), (5, 7), (3, 8), (3, 10)}	53	0,4966
8	{(4, 7), (2, 4), (4, 8), (6, 8), (4, 9), (5, 7), (1, 5), (3, 8), (3, 10)}	55	0,5681
9	{(4, 7), (2, 4), (4, 9), (5, 7), (6, 9), (6, 8), (1, 5), (3, 8), (3, 10)}	55	0,6449
10	{(6, 8), (6, 9), (4, 9), (2, 4), (2, 7), (5, 7), (1, 5), (3, 8), (3, 10)}	56	0,7027
11	{(2, 4), (2, 7), (4, 9), (5, 7), (6, 9), (1, 5), (1, 8), (3, 8), (3, 10)}	61	0,7271
12	{(2, 4), (2, 7), (4, 9), (5, 7), (6, 9), (1, 5), (3, 7), (3, 10), (3, 8)}	61	0,7468
13	{(2, 7), (5, 7), (1, 5), (2, 9), (4, 9), (6, 9), (3, 7), (3, 10), (3, 8)}	63	0,8180

Tabela 6.3: Conjunto de soluções, seus custos e valores de pertinência

Se aplicarmos o algoritmo para obter a solução de maior confiabilidade, teremos como $\beta = \{0.9501, 0.9568, 0.8939, 0.9318, 0.8381, 0.9797, 0.8998, 0.8385, 0.9797, 0.9883\}$, portanto $\alpha = 0,8381$. Para este valor, existem três componentes: a formada pelos nós 1 e 5; a formada pelos nós 2, 4, 6, 7 e 9 e a componente formada pelos nós 3, 8 e 10. Observando as arestas que podem conectar estas componentes, temos a aresta (3,7) com valor de pertinência igual a 0,8216. Colocando as arestas que possuem valor de pertinência entre 0,8216 e 0,8381 obtemos um subgrafo que possui duas componentes conectadas, uma formada pelos nós 1 e 5 e a outra com os nós restantes. Novamente, encontramos a aresta (5,7) com pertinência 0,8180. Agregando as arestas com valores de pertinência entre 0,8180 e 0,8216 obtemos um subgrafo conectado que é a própria árvore geradora mínima com pertinência igual a 0,8180.

Considerando a heurística CH96a, basta saber o valor de confiabilidade escolhido pelo decisor e efetuar uma iteração do algoritmo DVV90AGM para este valor de corte. Para o CH96b, considerando os valores da Tabela 3, temos que a árvore 13 é a escolhida pela heurística com que possui a melhor relação entre confiabilidade e custo. Para a heurística proposta neste trabalho, a partição depende do decisor. Se for escolhido $h = 3$, teremos um subconjunto com quatro árvores geradoras. Para o problema das Tabelas C.3, C.4 e 6.3 seriam as árvores 1, 4, 8 e 13 da Tabela 6.3; com seus respectivos custos e valores de pertinência. Para partições maiores,

o subconjunto será mais bem representado.

Para um problema maior, com um grafo completo de 100 nós e 4.950 arestas, o conjunto será consideravelmente maior. Gerando de forma aleatória os valores de pertinência e custos, como no problema de 10 nós anterior, obtemos um conjunto com cerca de 191 árvores geradoras mínimas. Os valores de pertinência das árvores do conjunto variam entre 0,0016 e 0,9580. Neste caso justifica-se o uso de heurísticas para redução da quantidade de informação ao decisor ou a obtenção apenas da árvore de maior valor de pertinência. Os dados de entrada e das soluções não devem ser apresentados pelo grande volume de informações.

As heurísticas e algoritmos tratados para o problema da árvore geradora mínima em um grafo fuzzy não devem ser comparados, pois cada um tem um objetivo diferente. É interessante que o decisor tenha todos disponíveis para utilizar aquele que for mais conveniente para o tipo e tamanho de problema a ser resolvido.

Capítulo 7

Problemas de emparelhamento fuzzy: designação e casamento estável

Os problemas de emparelhamento possuem um conjunto variado de aplicações (ver Seção 2 do Capítulo 12 de (Ahuja et al., 1993)) e normalmente são problemas combinatoriais que podem ser resolvidos considerando sua estrutura de grafos. Os exemplos mais comuns envolvem a melhor associação entre pessoa/trabalho, máquina/tarefa, profissional/horário de trabalho dentre diversos outros problemas nos quais necessitem associar pares de objetos de dois conjuntos distintos.

Neste trabalho, vamos estudar o problema de designação fuzzy e o problema do casamento fuzzy. O problema de designação será estudado para o caso em que a estrutura de grafos é crisp e os parâmetros são fuzzy e o caso da estrutura de grafos fuzzy. No caso do casamento estável será apenas apresentado o problema com estrutura de grafos fuzzy estudado por (Nair, 2000).

7.1 Definição do caso clássico

Dado um grafo $G = (N, A)$, um emparelhamento é um subgrafo com a propriedade de que dois arcos nunca são incidentes no mesmo nó. Sujeito a certas restrições, o problema é encontrar uma combinação factível que otimize algum objetivo como, por exemplo, minimizar o custo total dos arcos na solução.

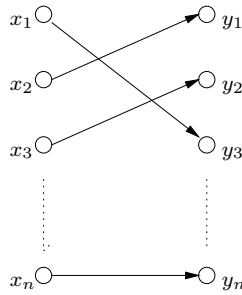


Figura 7.1: Exemplo de um problema de emparelhamento

O *problema de designação* clássico é um problema de emparelhamento perfeito, onde o grafo é bipartido (isto é, existem dois conjuntos de nós, N_1 e N_2 com $|N_1| = |N_2| = m$, $N_1 \cap N_2 = \emptyset$ e $N_1 \cup N_2 = N$) e a combinação deve ser de tal forma que cada nó em N_1 tenha associado exatamente um nó em N_2 . O custo desta combinação deve ser mínimo.

Outro tipo de emparelhamento, chamado de *problema do emparelhamento máximo* (ou segundo Ahuja et al. (1993), *cardinality problem*) corresponde a encontrar o emparelhamento com o maior número de nós.

Outra variante do problema do emparelhamento é o *problema do casamento estável*. Neste caso, temos subconjuntos N_1 e N_2 , representando os conjuntos de homens e mulheres, onde cada homem tem uma lista de preferência dentre o grupo de mulheres e as mulheres também tem suas listas de preferência dentre o grupo de homens. Um emparelhamento factível (ou um casamento estável) dos dois grupos ocorre quando nenhum par prefere os parceiros do outro casal ao invés do atual e vice-versa. O objetivo é encontrar o casamento estável entre os dois grupos.

Para estes problemas crisp, foram desenvolvidos algoritmos que utilizam a estrutura de grafos e resolvem estes problemas combinatoriais em tempo computacional factível, como por exemplo, o Método Húngaro (Ahuja et al., 1993).

7.1.1 Estudos encontrados na literatura sobre os problemas de emparelhamento fuzzy

(Herrera et al., 1994) foi um dos primeiros trabalhos encontrados para o problema de designação cujos parâmetros apresentam incertezas. A proposta foi de tentar resolver o problema por meio de algoritmos genéticos (Apêndice B) utilizando variáveis lingüísticas (Capítulo 2). Neste caso, o trabalho foi conceitual, não apresentando implementações ou exemplos.

Nair (2000) resolve o problema do casamento estável fuzzy. Considera que o arco que liga uma “mulher” de N_1 a um “homem” de N_2 possui um valor de pertinência associado, representando o interesse de uma mulher por aquele pretendente. O problema é encontrar o casamento estável de maior grau de pertinência.

Em (Lin e Wen, 2004), o problema de designação com parâmetros fuzzy é tratado por meio de programação linear *fracionária*. Neste trabalho, os números fuzzy associados podem ser discretos e não-normalizados. O objetivo é encontrar uma solução que maximize a função que descreve o grau de satisfação do decisor, baseado no critério de Bellman e Zadeh (Seção 4.1.1).

Em (Bershtein e Dziouba, 2000), o objetivo é encontrar em um grafo fuzzy a melhor partição do conjunto de nós N . Este trabalho não será analisado, pois assumiremos que o grafo bipartido é dado.

Nas próximas seções, serão apresentados os estudos realizados neste trabalho sobre o problema de designação com parâmetros fuzzy (Seção 7.2) e o problema de designação em um grafo fuzzy (Seção 7.3). Na Seção 7.4 será apresentado o problema do casamento estável proposto por (Nair, 2000).

7.2 Caso I: Problema de designação com estrutura do grafo crisp e custo fuzzy

Dado um grafo bipartido $G = (N, A) = (N_1, N_2; A)$, temos que o problema da designação clássico pode ser formulado como:

$$\begin{aligned} \text{Min } z &= \sum_{(i,j) \in A} c_{ij} x_{ij} \\ \text{s.a. } \sum_{j \in N_2} x_{ij} &= 1 & i \in N_1, (i, j) \in A \\ \sum_{i \in N_1} x_{ij} &= 1 & j \in N_2, (i, j) \in A \\ x_{ij} &\in \{0, 1\} & (i, j) \in A \end{aligned}$$

onde

$$x_{ij} = \begin{cases} 1, & \text{o emparelhamento é feito} \\ 0, & \text{caso contrário} \end{cases}$$

O objetivo é minimizar o custo total z , sujeito à restrição de que cada nó $i \in N_1$ esteja designado a apenas um outro nó $j \in N_2$.

Neste primeiro estudo, os custos associados ao problema são números fuzzy \tilde{c}_{ij} , $(i, j) \in A$. O grafo é clássico, isto é, o valor de pertinência associado aos nós e arcos é unitário.

Este problema pode ser baseado no mesmo procedimento para a construção do algoritmo para o problema da árvore geradora mínima com parâmetros fuzzy (Capítulo 6, Seção 6.2). No caso, temos que a designação com maior grau de possibilidade também é encontrada através da solução do problema crisp associado utilizando o ínfimo do valor modal do custo fuzzy como custo do problema crisp baseado na Proposição 2.7.1 e no Teorema 2.7.1.

Para gerar as outras soluções com seus graus de possibilidade seguiremos os mesmos passos do problema da árvore geradora mínima, com a diferença que a verificação da factibilidade da designação é muito mais fácil do que no problema da árvore geradora mínima, como podemos ver na Figura 7.2:

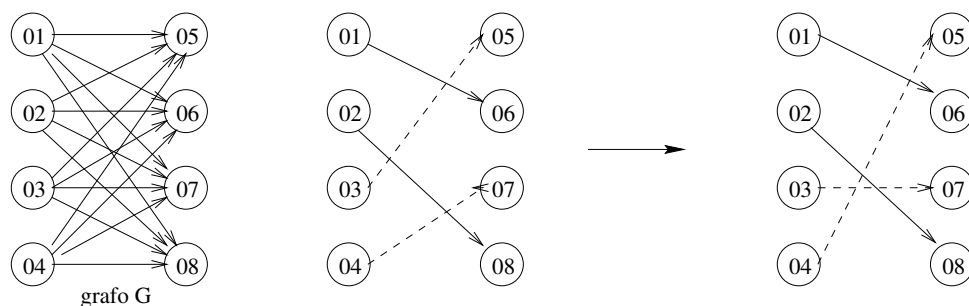


Figura 7.2: Exemplo de construção de soluções para o problema

Para este problema é necessário apenas verificar a existência ou não de um determinado arco. Não existe a necessidade de verificar a conectividade do subgrafo. Com isso, podemos adaptar o algoritmo de árvore geradora mínima no algoritmo dado a seguir.

7.2.1 Algoritmo proposto para o problema de designação com parâmetros fuzzy: PD-PF

Procedimento 1:

Passo 0: (*Inicializar os parâmetros*) Encontre a solução do problema de designação H^* associada a solução do grafo crisp $G_C = (N, A)$ com custo $l_{ij} = \inf_{\mu_{ij}(u)=1} \{u\}$. Faça $\delta = \sup\{supp\{\tilde{l}_{H^*}\}\}$.

Passo 1: Para $k = 1$ até $k = |N| - 2$, faça:

- Para cada k arcos da solução H^* (estes k arcos $\in R'$), execute o procedimento 2.

Passo 2: Faça:

1. $A_1 \leftarrow A - H^*$. Seja $G_1 = (N, A_1)$.
2. Encontre H_1 através da resolução do grafo crisp $\underline{G}_1 = (N, A_1)$ com $l_{ij} = \inf\{supp\{\tilde{l}_{ij}\}\}$.
3. Se não houver mais soluções ou se $\inf\{supp\{\tilde{l}_{H_1}\}\} > \delta$, então FIM. Caso contrário, para $k = 1$ até $|N| - 2$, faça:
 - Para cada k arcos de H_1 (estes k arcos $\in R'$), execute o procedimento 2.

Procedimento 2: (Construir as designações e calcular a possibilidade)

Passo 0: (Inicialização dos parâmetros) Sejam k , H e R' dados. Faça $\tilde{l}' \leftarrow \sum_{(i,j) \in R'} \tilde{c}_{ij}$.

Passo 1: (Comparação com os arcos do corte) Para cada k arcos $(s, t) \in [S, \bar{S}]$ (estes k arcos $\in R''$), faça:

- $\tilde{l}'' \leftarrow \sum_{(i,j) \in R''} \tilde{c}_{ij}$
- Se $\inf\{\tilde{l}_H - \tilde{l}' + \tilde{l}''\} > \delta$, então descarte esta designação. Caso contrário, $w \leftarrow Poss(l'' \leq l')$.
- Faça $D_{ij} \leftarrow \max\{D_{ij}, w\}$, $(i, j) \in R''$.

Passo 2: Caso não existam mais associações a serem formadas, volte ao procedimento 1.

7.2.2 Algoritmo genético para o problema de designação com parâmetros fuzzy

Como no problema da árvore geradora mínima, este problema é de difícil solução e o uso de heurísticas e/ou meta-heurísticas é atraente. Então, para este problema também será proposto um algoritmo genético (Apêndice B).

A seguir são descritos os componentes do nosso algoritmo genético:

Codificação: Para um grafo bipartido $G : (N_1, N_2, A)$, com $m = |N_1| = |N_2|$ e $n = |A|$, temos que o cromossomo possui m genes. O índice de cada gene é igual a um nó de N_1 e o valor contido neste gene é igual ao nó em N_2 . Pelo exemplo da Figura 7.3, o nó 1 é designado ao nó 6, o nó 2 ao nó 8, o nó 3 ao nó 5 e o nó 4 ao nó 7. A designação é válida desde que $(i, j) \in A$, para todos os arcos envolvidos. Se o grafo bipartido for completo, então não há necessidade de verificar a existência do arco.

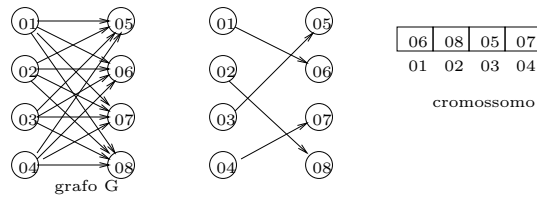


Figura 7.3: Codificação de uma solução

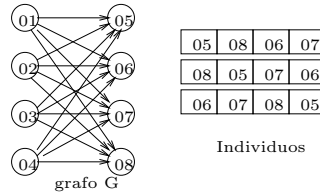


Figura 7.4: Inicialização

Inicialização: pode ser realizada utilizando permutações dos valores dos nós, desde que comprovem a existência dos arcos pertencentes ao indivíduo

Crossover: Para este caso, o crossover OX para o problema do caixeiro viajante é atraente (ver Apêndice B). Neste caso, temos dois indivíduos, P_1 e P_2 . Copia-se um trecho do indivíduo P_1 e completamos com a seqüencia dos nós do indivíduo P_2 , desde que existam os arcos. Senão procede-se nova tentativa de combinação dos dois indivíduos.

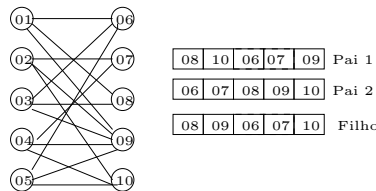


Figura 7.5: Crossover

Mutação: Neste caso, um ou mais arcos são proibidos e outra combinação é encontrada, aleatoriamente ou através do algoritmo Húngaro

Outras informações:

- A medida de adaptação, ou valor de fitness, é calculada como o valor de possibilidade da solução ser menor do que a melhor solução encontrada.
- O tamanho da população e o número de gerações deve ser testado para diferentes tamanhos de problema, levando em consideração a dimensão deste.

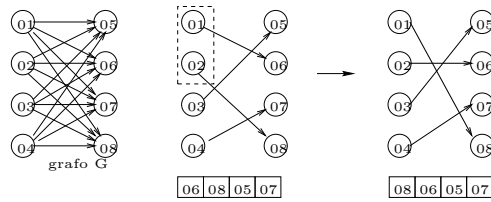


Figura 7.6: Mutaç o

- A seleç o dos indiv duos para reproduç o   feita de forma aleat ria, em uma tentativa de manter a diversidade da populaç o.
- A nova populaç o   formada pelos filhos de crossover e mutaç o, pelos melhores indiv duos e por indiv duos escolhidos de forma aleat ria.

Para o caso de um grafo bipartido completo, a abordagem proposta nesta seç o se comporta bem, pois todas as configuraç es poss veis do cromossomo s o vi veis. No caso do grafo n o possuir todos os arcos, ent o deve-se estudar outras codificaç es ou a criaç o de arcos artificiais. No segundo caso, adiciona-se arcos ao grafo para torn -lo completo, mas estes arcos possuem valores de custo (ou outro par metro) que o torna pouco atraente para a soluç o. No caso de um problema de minimizar os custos de designaç o, por exemplo, este arco artificial teria um valor muito superior aos dos arcos originais do grafo.

7.2.3 Exemplos e testes

Como primeiro exemplo, seja o grafo da Figura 7.2 com os seguintes custos fuzzy:

Em um problema com 04 pares de n s, temos 16 ($n!$) tipos de designaç es poss veis. Neste exemplo, 04 emparelhamentos tem um grau de possibilidade n o-nulo de ser a designaç o com menor custo. Utilizando o procedimento descrito na Seç o 7.2.1, a soluç o com maior grau de possibilidade, $D_1 = 1$, t m a designaç o $M_1 = \{(1, 8), (2, 7), (3, 6), (4, 5)\}$ com peso fuzzy (125, 150, 170). Usando um valor de corte igual a $\sigma = 170$, eliminamos os outros candidatos. As outras tr s designaç es com algum grau de possibilidade de ser a soluç o com menor peso s o: $M_2 = \{(1, 5), (2, 7), (3, 6), (4, 8)\}$ com grau de possibilidade igual a $D_2 = 0.8889$, $M_3 = \{(1, 8), (2, 6), (3, 7), (4, 5)\}$ com $D_3 = 0.8750$ e $M_4 = \{(1, 5), (2, 6), (3, 7), (4, 8)\}$ com grau de possibilidade igual a $D_4 = 0.7500$. Para um problema pequeno como este, o algoritmo gen tico encontrou o conjunto soluç o em todas as tentativas.

Para testar tamb m o algoritmo gen tico, recorreremos a um exemplo constru do aleatoriamente com 10 pares de n s. Os par metros fuzzy s o n meros fuzzy triangulares com 10% de

arcos	pesos	arcos	pesos
(1,5)	(40,45,50)	(3,5)	(80,90,100)
(1,6)	(100,105,110)	(3,6)	(50,60,65)
(1,7)	(95,100,105)	(3,7)	(20,25,35)
(1,8)	(25,30,35)	(3,8)	(70,75,80)
(2,5)	(45,50,55)	(4,5)	(40,45,50)
(2,6)	(50,55,60)	(4,6)	(80,90,100)
(2,7)	(10,15,20)	(4,7)	(90,95,100)
(2,8)	(95,100,105)	(4,8)	(30,35,40)

Tabela 7.1: Peso dos arcos

espalhamento em torno do valor modal. O problema possui $10! = 3.628.800$ possíveis emparelhamentos. Para problemas deste porte, construir todo o conjunto solução é uma tarefa muito difícil. O algoritmo genético foi testado para diferentes combinações dos parâmetros (tamanho da população, crossover, mutação e número de gerações). O que podemos observar é que o conjunto fuzzy de soluções é diretamente dependente destes parâmetros. Para ilustrar, vamos observar dois casos: o primeiro com a população de $2 \times n$ (n é o número de pares de nós) e 100 gerações e o segundo caso com n^2 indivíduos na população. As Figuras 7.7 e 7.8 mostram os valores de possibilidade da população ao longo da geração e o valor do melhor fitness e do fitness médio da população, respectivamente para o primeiro exemplo. Para o segundo exemplo, os mesmo dados estão disponíveis nas Figuras 7.9 and 7.10.

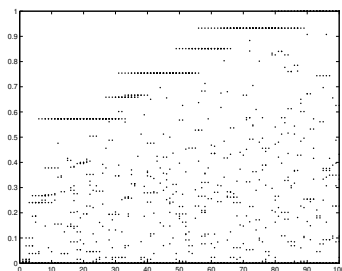


Figura 7.7: Valor de possibilidade da população no primeiro caso

O primeiro teste encontrou 450 designações diferentes com grau de possibilidade não nulo, dentre 820 verificados. O segundo teste avaliou 4014 designações e encontrou 2104 com grau de

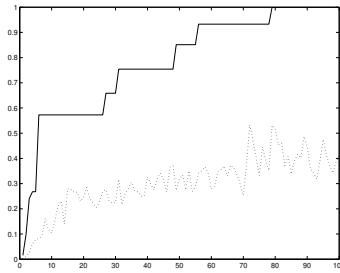


Figura 7.8: Melhor fitness e média da população para o primeiro caso

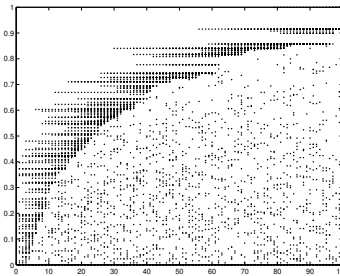


Figura 7.9: Valor de possibilidade da população para o segundo caso

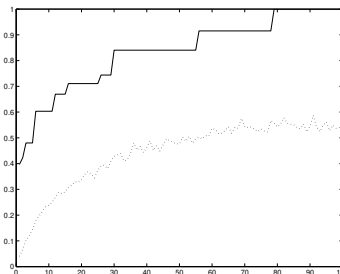


Figura 7.10: Melhor fitness e média da população para o segundo caso

possibilidade não nulo. Mantendo a diversidade da população, podemos obter mais indivíduos.

7.3 Estudo II: o problema da designação em um grafo fuzzy

O problema da designação com graus de pertinência associado a seus nós e arcos não tem o mesmo tratamento dos estudos vistos anteriormente. O algoritmo proposto na Seção 6.3 basicamente:

1. verifica a maior pertinência dos arcos ligados a um dado nó;
2. toma o menor dentre estes valores para fazer um α -corte gerando um grafo G' ;

3. verifica se o grafo resultante é conexo;
4. resolve o problema crisp associado.

Neste tipo de problema, não é necessário verificar a conectividade do grafo resultante, pois a escolha do nível de corte α garante as conexões necessárias para a resolução do problema. Se houvesse um nó, p , sem arco de conexão com o grafo G' resultante, então os arcos que incidem sobre p deveriam possuir grau de pertinência menor que o corte escolhido, o que não é possível pois α -corte é escolhido como o menor valor entre maiores pertinências de cada nó. O que deve ser observado é que apenas a determinação do α -corte não garante encontrar um problema crisp associado que tenha solução como no caso do problema da árvore geradora mínima e problema do caminho mínimo .

O α -corte garante que cada nó de N_1 e cada nó de N_2 tenham arcos incidentes. Mas os emparelhamentos entre os nós pode não ser factível. Por exemplo, na Figura 7.11:

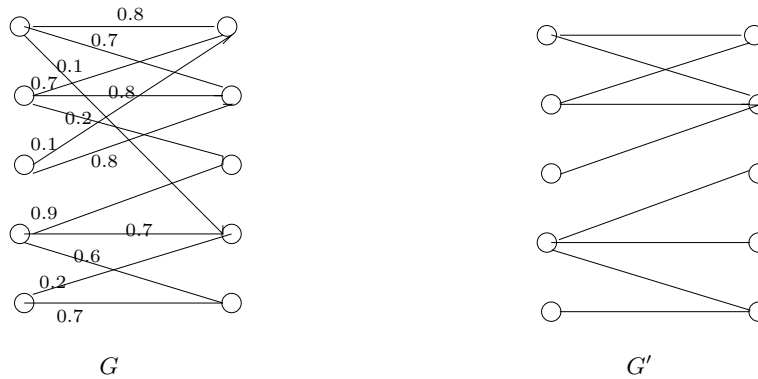


Figura 7.11: Exemplo de um grafo bipartido e o grafo G' resultante do α -corte ($\alpha = 0.6$)

Um grupo dos três primeiros nós de N_1 tem somente os mesmos dois nós de N_2 para o emparelhamento, logo não existe solução para este grafo.

Considerando o problema do casamento estável e o estudo de (Nair, 2000), pois o problema de designação é um emparelhamento completo, podemos nos valer do Teorema 7.4.1, apresentado na Seção 7.4, para propor o seguinte algoritmo:

Algoritmo de Delgado et. al, (1990) adaptado para o problema de designação: PD-DVV90

Passo 1: Faça

$$\beta_i \leftarrow \max_{j \in N_2} \{\mu_{ij}\}; \quad i \in N_1$$

$$\gamma_j \leftarrow \max_{i \in N_1} \{\mu_{ij}\}; \quad j \in N_2$$

$$\alpha = \min\{\min_{i \in N_1}\{\beta_i\}, \min_{j \in N_2}\{\gamma_j\}\}$$

$\bar{\alpha} \leftarrow 1$ e $A' \leftarrow \emptyset$.

Passo 2: Construa um grafo G' sendo

$$N' \leftarrow N$$

$$A' \leftarrow A' \cup \{(i, j) \in A \mid \alpha \leq \mu_{ij} \leq \bar{\alpha}\}$$

vá ao passo 3.

Passo 3: Caso exista algum k , $1 \leq k \leq n$ tal que um dado conjunto de k nós de N_1 incida em um conjunto menor do que k nós de N_2 (Teorema 7.4.1), então faça

$$\beta_i \leftarrow \max_{j \in N_2 \mid \mu_{ij} < \alpha} \{\mu_{ij}\}; \quad i \in N_1$$

$$\gamma_j \leftarrow \max_{i \in N_1 \mid \mu_{ij} < \alpha} \{\mu_{ij}\}; \quad j \in N_2$$

$$\alpha = \min\{\min_{i \in N_1}\{\beta_i\}, \min_{j \in N_2}\{\gamma_j\}\}$$

$\bar{\alpha} \leftarrow \alpha$ e volte ao passo 2. Caso contrário, vá ao passo 4.

Passo 4: Resolva o problema de designação para G' por meio de um algoritmo clássico.

O passo 1 garante que todos os nós, de N_1 e N_2 terão um arco associado. O passo 2 constrói o grafo referente a solução com maior pertinência. O passo 3 verifica se o grafo construído no passo 2 têm solução e o passo 4 resolve o problema crisp referente ao grafo G' .

Novamente, este algoritmo é interessante para saber a solução de maior pertinência em grafos com uma variação grande de valores de pertinência e com um número relativamente grande de arcos.

7.4 Problema do casamento estável

O problema de casamento estável corresponde a seguinte pergunta: *se existe um conjunto de mulheres, cada qual conhecendo alguns homens, sob quais condições todas as mulheres poderiam casar com um homem que elas conhecem?*

Como (Nair, 2000) cita em seu trabalho, este problema pode ser visto como um problema de emparelhamento completo dos conjuntos de N_1 mulheres e de N_2 homens, cuja condição é apresentada no seguinte Teorema.

Teorema 7.4.1. *(Condição de Hall) Uma condição necessária e suficiente para uma solução do problema de casamento estável envolvendo n garotas é que cada conjunto de k garotas coletivamente conheçam pelo menos k garotos para $1 \leq k \leq n$.*

Aqui apresentaremos a teoria e o algoritmo proposto por (Nair, 2000) para o caso do problema do casamento estável fuzzy: se existe um conjunto finito de garotas, sendo que cada uma gosta de vários garotos com vários graus de escolha, sob quais condições poderiam todas as garotas casar com os garotos que elas mais gostam?

Para tal problema, Nair propôs as seguintes definições:

Definição 7.4.1. *Um grafo fuzzy $G : (N, \sigma, \mu)$ é um grafo bipartido fuzzy se o conjunto de nós N é uma união disjunta dos conjuntos N_1 e N_2 tal que, $\forall x \in N_1, \forall y \in N_2, \mu_{xy} \geq 0$ e $\forall x \in N_1, \forall y \in N_2, \mu_{yx} = 0$.*

Usando $G = (N_1, N_2, \sigma, \mu)$ para representar um grafo bipartido fuzzy, temos:

Definição 7.4.2. *Seja $G = (N_1, N_2, \sigma, \mu)$ um grafo bipartido fuzzy. Um emparelhamento completo fuzzy de N_1 para N_2 é uma função bijetora $f : N_1 \rightarrow N_2$ tal que $\mu(x, f(x)) > 0, \forall x \in N_1$.*

Definição 7.4.3. *Seja $G = (N_1, N_2, \sigma, \mu)$ um grafo bipartido fuzzy. Um α -emparelhamento completo fuzzy de N_1 para N_2 é uma função bijetora $f : N_1 \rightarrow N_2$ tal que $\mu(x, f(x)) \geq \alpha, \forall x \in N_1$.*

Proposição 7.4.1. *Seja $G = (N_1, N_2, \sigma, \mu)$ um grafo bipartido fuzzy. Existe um α -emparelhamento completo fuzzy de N_1 para N_2 se, e somente se, G^α tem um emparelhamento completo de N_1 para N_2 .*

Isto nos leva à seguinte Proposição:

Proposição 7.4.2. *Seja $G = (N_1, N_2, \sigma, \mu)$ um grafo bipartido fuzzy. Existe um emparelhamento completo fuzzy de N_1 para N_2 se, e somente se existe um $\alpha > 0$ tal que G^α tem um emparelhamento completo de N_1 para N_2 .*

Considerando que a solução do problema de máximo emparelhamento completo fuzzy tem fator $\bar{\alpha}$, tal que para todos os α 's que fornecem solução ao problema $\alpha \leq \bar{\alpha}$, temos:

Teorema 7.4.2. *Seja $G = (N_1, N_2, \sigma, \mu)$ um grafo bipartido fuzzy. Então são equivalentes*

1. *Existe um emparelhamento completo fuzzy de N_1 para N_2 em G .*

2. Existe um máximo emparelhamento completo de N_1 para N_2 .
3. Existe $\alpha > 0$ tal que G^α satisfaz a condição necessária e suficiente do Teorema 7.4.1.

(Nair, 2000) propõe um algoritmo para encontrar o máximo emparelhamento completo fuzzy que satisfaz as condições do problema.

Algoritmo (Nair, 2000):

Entrada: n : número de “mulheres”;

m : número de “homens”;

M : matriz fuzzy sendo M_{ij} o valor associado pela garota i ao homem j .

Saída: α : máximo fator de emparelhamento completo. Se $\alpha = 0$, então o problema não tem solução.

f : emparelhamento completo máximo

Passo 1: Ordene os valores encontrados na matriz M . Seja $\alpha_1 < \alpha_2 < \dots < \alpha_r$. Faça $\alpha_0 = 0$.

Passo 2: Encontre $\alpha_k = \min_i \{ \max_j \{ M_{ij} \} \}$

Passo 3: $\alpha \leftarrow \alpha_k$

Enquanto $\alpha > 0$

Se a condição de Hall (Teorema 7.4.1) é falsa para G^α então

$k \leftarrow k - 1$

$\alpha \leftarrow \alpha_k$

senão $\alpha = 0$ **Fim.**

fim do enquanto.

Se $\alpha = 0$ Fim do algoritmo.

Passo 4: Para cada i faça

$B[i] \leftarrow \{ j | M[i, j] \geq \alpha \}$

Seja $I = \{ 1, 2, \dots, n \}$

Enquanto $I \neq \emptyset$

Escolha o i com menor número de elementos

Seja $j \in B[i]$ tal que M_{ij} é máximo na linha i .
 Faça $f(i) \leftarrow j$
 $I = I - \{i\}$

Para cada $i \in I$ faça
 $B[i] \leftarrow B[i] - j$

fim do enquanto.

7.4.1 Exemplo numérico

Seja um problema com 5 meninas (A1 a A5) e 6 meninos (B1 a B6). A matriz M é dada abaixo:

$$M = \begin{array}{c|cccccc} & \text{B1} & \text{B2} & \text{B3} & \text{B4} & \text{B5} & \text{B6} \\ \hline \text{A1} & 0,1 & 0,2 & 0,3 & 0,5 & 0 & 1 \\ \text{A2} & 0,5 & 0 & 0 & 0,6 & 0,7 & 0,4 \\ \text{A3} & 0,4 & 0,6 & 0,7 & 0,6 & 0 & 0 \\ \text{A4} & 0,5 & 0 & 0 & 0 & 0,9 & 1 \\ \text{A5} & 0 & 0,3 & 0,7 & 0,4 & 0,2 & 0,4 \end{array}$$

O menor valor não-nulo contido em M é igual a 0,1 (elemento m_{11}). Considerando $G_{0,1}$:

$$\begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

É fácil verificar que o Teorema 1 confirma a existência de um emparelhamento completo em $G_{0,1}$. Para encontrar o máximo emparelhamento completo em G , calculamos: $\min_i \{\max_j \{M_{ij}\}\} = \min\{1; 0,7; 0,7; 1; 0,7\} = 0,7$ Para $G_{0,7}$:

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}$$

Neste caso, não existe um emparelhamento completo. Para as mulheres A1, A2 e A4 existem somente dois homens disponíveis: B5 e B6. Então, calcula-se o conjunto de corte para o valor de pertinência menor: 0,6.

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}$$

Para o valor de pertinência igual a 0,6 existe um emparelhamento completo: A1 com B6, A5 com B3, A4 com B5, A2 com B4 e A3 com B2. Este é o máximo emparelhamento completo.

Capítulo 8

Fluxos Máximo Fuzzy

Uma rede é um grafo direcionado, capacitado e sem laços onde existe fluxo para atender as necessidades dos nós. Associando a cada arco (i, j) da rede um valor $u(i, j)$ que corresponde à capacidade do arco, podemos definir uma função $\phi(i, j)$ que corresponde ao fluxo do arco. Esta função satisfaz às seguintes restrições:

1. $\phi(i, j) \geq 0$, isto é, o fluxo é não negativo em cada arco;
2. $\phi(i, j) \leq u(i, j)$: fluxo não excede a capacidade do arco;
3. satisfaz a regra de conservação de fluxo, ou seja, o fluxo que entra em um nó mais o localmente gerado é o mesmo que sai mais o localmente consumido.

Os objetivos de um problema de fluxos em redes podem ser:

- obter o fluxo máximo ϕ_0 da rede;
- obter o fluxo de custo mínimo na rede, $C = \sum_{(i,j) \in A} c_{ij}x_{ij}$, de modo a atender a uma demanda pré-determinada entre dois nós, etc.

Outras definições e a teoria envolvida relativa ao assunto podem ser vistas em (Bazaraa et al., 1990) e (Ahuja et al., 1993).

O problema com estrutura crisp e capacidade fuzzy leva em consideração o fluxo que o decisor considera aceitável e a tolerância à violação deste fluxo. Neste trabalho pretende-se apresentar o conjunto de solução do problema do fluxo máximo com parâmetros fuzzy e fluxo contínuo sem que o decisor necessite definir o fluxo aceitável previamente.

A seguir, vamos apresentar a teoria e os algoritmos propostos para os casos discutidos do problema do fluxo máximo fuzzy. Na Seção 8.1 é feita uma revisão bibliográfica sobre este tópico. Na Seção 8.2 é apresentada a definição do problema de fluxo máximo com capacidade fuzzy. Serão apresentadas as definições e o algoritmo proposto por (Chanas e Kolodziejczyk, 1984) e (Chanas e Kolodziejczyk, 1986) para o problema considerando a incerteza e a informação do decisor. Na Seção 8.2.6 será apresentada uma nova abordagem do problema, um novo algoritmo exato e um algoritmo aproximado. Finalmente, na Seção 8.2.8 será apresentada o algoritmo proposto por (Chanas et al., 1995) para o problema de fluxo máximo com a estrutura do grafo fuzzy.

8.1 Revisão Bibliográfica

(Kim e Roush, 1982) desenvolvem a teoria de fluxo fuzzy, apresentando condições necessárias e suficientes para um fluxo ótimo, por meio de definições sobre matrizes fuzzy. O problema tratado possui estrutura crisp e, a cada arco da rede, os limitantes inferior e superior do fluxo na rede são números fuzzy.

Em (Chanas e Kolodziejczyk, 1982), (Chanas e Kolodziejczyk, 1984) e (Chanas e Kolodziejczyk, 1986), os autores desenvolvem os conceitos de fluxo fuzzy e fluxo máximo fuzzy. Em (Chanas e Kolodziejczyk, 1982) o grafo tem estrutura crisp e os arcos têm capacidade fuzzy, isto é, os arcos têm uma função de pertinência associada ao fluxo, dependente da capacidade. Em (Chanas e Kolodziejczyk, 1984), o problema do fluxo máximo é novamente abordado para o caso do fluxo ser um número real fuzzy. As capacidades inferior e superior têm pertinências associadas. A teoria foi desenvolvida para o fluxo inteiro fuzzy em (Chanas e Kolodziejczyk, 1986), com suporte teórico necessário e exemplos do algoritmo proposto.

(Peng e Juang, 1993) trabalha com um problema onde a cada arco estão associadas uma capacidade (crisp) e uma pertinência do fluxo. O trabalho propõe um algoritmo para o problema do fluxo máximo em grafos com estrutura fuzzy, onde o objetivo é encontrar a máxima pertinência associada a um fluxo v . Apresenta um exemplo do algoritmo.

Em (Chanas et al., 1995) os trabalhos desenvolvidos em (Chanas e Kolodziejczyk, 1982), (Chanas e Kolodziejczyk, 1984), (Chanas e Kolodziejczyk, 1986) e (Delgado et al., 1985) são reunidos para definição do problema de fluxo fuzzy, rede fuzzy e o problema de transporte fuzzy.

O Capítulo 3 do livro de (Malik e Moderson, 2001) foi totalmente dedicado à teoria de rede fuzzy, ao problema do fluxo máximo fuzzy e redes de Petri fuzzy. Os artigos citados acima sobre fluxo máximo estão presentes no livro e são também as referências em relação ao problema de

fluxo máximo fuzzy (inteiro e real) e as definições sobre fluxo (aceitável) fuzzy.

8.2 Problema do Fluxo Máximo Fuzzy

Como vimos na Seção 8.1, quase todas as referências sobre a teoria de fluxos em redes fuzzy, propõem algoritmos para a resolução do problema do fluxo máximo fuzzy. Aqui vamos apresentar parte relevante da teoria já formulada e os algoritmos propostos.

8.2.1 Definição do caso clássico

Seja uma rede com m nós e n arcos através do qual o fluxo de um determinado objeto deverá passar. Cada arco deverá ter um limitante inferior (normalmente, $l_{ij} = 0$) e superior u_{ij} na sua capacidade. Não há custos envolvidos no fluxo. O objetivo deste problema é determinar qual o fluxo máximo v_0 que pode passar de um nó s ao nó t da rede.

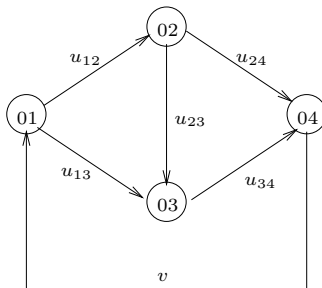


Figura 8.1: Exemplo de um problema do fluxo máximo

O problema do fluxo máximo pode ser modelado como

$$\begin{aligned}
 & \text{Max} && v \\
 & \text{s.a.} && \sum_{j:(i,j) \in A} x_{ij} - \sum_{j:(j,i) \in A} x_{ji} = \begin{cases} v, & i = s \\ 0, & \forall i \in N - \{s, t\} \\ -v & i = t \end{cases} \\
 & && 0 \leq x_{ij} \leq u_{ij} \quad (i, j) \in A
 \end{aligned} \tag{8.1}$$

8.2.2 Algumas definições do caso do problema com capacidade fuzzy

Podemos associar uma função de satisfação ao atendimento de um certo fluxo

$$\mu_S(v) = \begin{cases} 1 & \text{se } v > v_0 \\ L(v_0, v_1; v) & \text{se } v_1 \leq v \leq v_0 \\ 0 & \text{se } v < v_1 \end{cases} \quad (8.2)$$

sendo $L(v_0, v_1; v)$ uma função linear tal que $L(v_0, v_1; v_0) = 1$ e $L(v_0, v_1; v_1) = 0$. O valor de $\mu_S(v)$ mostra o grau de satisfação do usuário com o fluxo v .

Cada restrição de capacidade $x_{ij}^v \leq u_{ij}$ pode ser associada a um conjunto fuzzy $C(i, j)$ de função de pertinência:

$$\mu_S(x_v) = \begin{cases} 1; & \text{se } x_{ij}^v < u_{ij} \\ \bar{L}(u_{ij}, \bar{u}_{ij}x_{ij}^v); & \text{se } u_{ij} \leq x_{ij}^v \leq \bar{u}_{ij} \\ 0 & \text{se } x_{ij}^v > \bar{u}_{ij} \end{cases} \quad (8.3)$$

onde $\mu_{ij}(x_v)$ fornece o grau de confiança da satisfação da restrição $x_{ij}^v \leq u_{ij}$.

O problema pode ser reduzido a encontrar um fluxo que maximize o valor da função de pertinência $\mu_D(x_v)$ em uma decisão fuzzy da forma de (Bellman e Zadeh, 1970) (vide 4.1.1) D , onde:

$$D = \bigcap_{(i,j) \in A} C_{ij} \bigcap S \quad (8.4)$$

$$\mu_D(x_v) = \mu_C(x_v) \bigwedge \mu_S(x_v) \quad (8.5)$$

$$\mu_C(x_v) = \bigwedge_{(i,j) \in A} \mu_{ij}(x_{ij}) \quad (8.6)$$

$\mu_C(x_v)$ denota o grau de satisfação simultânea das restrições de capacidade em todos os arcos da rede e μ_S o grau de satisfação do objetivo dado pelo decisor.

Definição 8.2.1. Um fluxo x_v^* é chamado de fluxo máximo se

$$x_v^* = \max_{x_v} \mu_D(x_v)$$

Definição 8.2.2. Um fluxo x_v com o maior v dentre outros fluxos máximos é chamado de fluxo ótimo.

Lema 8.2.1. Se x_v é um fluxo ótimo na rede G com capacidade C , $C(i, j) = u_{ij}$, então para qualquer caminho p do nó origem s ao destino t

- (i) existe um arco $(i, j) \in p$ tal que $\mu_{ij}(x_{ij}) = \mu_D(x_v)$
- (ii) existe um arco reverso $(j, i) \in p$ para o qual $x_{ji} = 0$.

Teorema 8.2.1. *Se x_v é um fluxo ótimo na rede G com capacidade C , existe um corte (X, \bar{X}) na rede satisfazendo as seguintes condições:*

- (i) $(i, j) \in A, i \in X, j \in \bar{X} \Rightarrow \mu_{ij}(x_{ij}) = \mu_D(x_v)$,
- (ii) $(i, j) \in A, i \in \bar{X}, j \in X \Rightarrow \mu_{ij}(x_{ij}) = 0$

8.2.3 Grafo crisp e capacidade do arco fuzzy: fluxo máximo com valores inteiros

Nesta parte, será discutido o problema de fluxos em redes com valores inteiros. Discussões mais detalhadas podem ser vistas em (Chanas e Kolodziejczyk, 1986) e (Malik e Moderson, 2001).

Lema 8.2.2. *Se x_v é o fluxo ótimo, então para qualquer caminho de s para t temos:*

- (i) existe um arco (i, j) tal que $\mu_{ij}(x_{ij} + 1) < \mu_D(x_v)$ ou
- (ii) existe um arco $(j, i) \in \overleftarrow{p}$ para o qual $x_{ji} = 0$

onde \vec{p} e \overleftarrow{p} são conjuntos de arcos *forward* e *backward* no caminho p , respectivamente.

Teorema 8.2.2. *Se x_v é um fluxo ótimo, então existe na rede um corte (X, \bar{X}) tal que as seguintes propriedades valem:*

- (i) $(i, j) \in A, i \in X, j \in \bar{X} \Rightarrow \mu_{ij}(x_{ij} + 1) < \mu_D(x_v)$
- (ii) $(i, j) \in A, i \in \bar{X}, j \in X \Rightarrow x_{ij} = 0$

Lema 8.2.3. *As seguintes relações valem:*

$$\bigvee_{x_v} \mu_D(x_v) = \bigvee_{z \in \mathcal{N}} \{\mu_C(\bar{x}_z) \bigwedge \mu_S(z)\}$$

onde $\mu_C(\bar{x}_z) = \bigvee_{x_z \in V(z)} \mu_C(x_z)$ e $V(z) = \{x_v | v = z\}$.

Segue do Lema 8.2.3 que para encontrar um fluxo ótimo, é suficiente considerar cada $V(z)$, $z \in \mathcal{N}$, somente para o fluxo $\bar{x}_z \in V(z)$ que satisfaz as condições da melhor forma.

Algoritmo (Chanas e Kolodziejczyk, 1986):

Passo 1: Encontre o fluxo máximo x_w no sentido crisp na rede S assumindo capacidades dos arcos iguais a u_{ij} , $(i, j) \in A$. Seja $\mu_C(x_w) = 1$. Se $\mu_S(v) = 1$, então x_w é ótimo FIM. Caso contrário vá ao passo 2.

Passo 2: Seja x_v um fluxo corrente. Determine um caminho p entre s e t e maximizando o valor da expressão

$$\mu^p(x_v) = \left(\bigwedge_{(i,j) \in \vec{p}} \mu_{ij}(x_{ij} + 1) \right) \bigwedge \left(\bigwedge_{(i,j) \in \overleftarrow{p}} \mu_{ij}(x_{ij} - 1) \right) \rightarrow \max$$

onde \vec{p} e \overleftarrow{p} são conjuntos de arcos *forward* e *backward* no caminho p , respectivamente. Assumindo que se $x_{ij} = 0$, então $\mu_{ij}(x_{ij} - 1) = -1$. Se $\mu_D(x_v) > \mu^p(x_v)$ então x_v é ótimo, FIM. Caso contrário vá ao passo 3.

Passo 3: Faça $x_v \leftarrow x_{v+1}$. x_{v+1} é o fluxo obtido pela adição de uma unidade no fluxo x_v no caminho p (adicionando uma unidade dos arcos *forward* e subtraindo uma unidade dos arcos *backward*). Volte ao passo 2.

8.2.4 Grafo crisp e capacidade do arco fuzzy: fluxo inteiro com restrições inferior e superior de capacidade

Cada restrição de capacidade $l_{ij} \leq x_{ij}^v \leq u_{ij}$ pode ser associado com um conjunto fuzzy $C(i, j)$ de função de pertinência:

$$\mu_R(x_v) = \begin{cases} 0 & \text{se } x_{ij} < \underline{l}_{ij} \\ L_l(\underline{l}_{ij}, l_{ij}, x_{ij}) & \text{se } \underline{l}_{ij} \leq x_{ij}^v < l_{ij} \\ 1 & \text{se } l_{ij} \leq x_{ij}^v < u_{ij} \\ L_r(u_{ij}, \bar{u}_{ij}, x_{ij}^v) & \text{se } u_{ij} \geq x_{ij}^v \geq \bar{u}_{ij} \\ 0 & \text{se } x_{ij}^v > \bar{u}_{ij} \end{cases} \quad (8.7)$$

onde $\underline{l}_{ij} < l_{ij}$ e $\bar{u}_{ij} > u_{ij}$ são as tolerâncias admissíveis para as capacidades inferior e superior, respectivamente.

Como no problema anterior, este pode ser reduzido a encontrar um fluxo que maximize o valor de pertinência $\mu_D(x_v)$ em uma decisão fuzzy, como nas equações 8.4, 8.5 e 8.6, sendo $\mu_{ij}(x_{ij})$ é da forma 8.7.

As propriedades estabelecidas podem ser vistas em (Chanas e Kolodziejczyk, 1986) e (Malik e Moderson, 2001). Abaixo apresentamos o algoritmo desenvolvido para este caso.

Algoritmo (Chanas e Kolodziejczyk, 1986):

Passo 1: Determine o fluxo máximo x_v^1 na rede assumindo fluxo nos arcos $x_{ij} \in [l_{ij}, \bar{u}_{ij}]$. Substitua $\bar{x}_v^1 \leftarrow x_v^1$ e vá ao passo 2.

Passo 2: Determine o caminho p da origem s ao destino t com respeito a \bar{x}_v^1 . Gere o fluxo x_{v+1}^2 adicionando uma unidade no fluxo x_v^1 no caminho p . Se $\mu_C(\bar{x}_v^1) \geq \mu_C^p(\bar{x}_v^1)$, então faça $\bar{x}_{v+1}^2 \leftarrow x_{v+1}^2$ e vá ao passo 5. Caso contrário vá ao passo 3.

Passo 3: Determine na rede um arco (k, r) para o qual $\mu_{kr}(x_{kr}^2) = \mu_C(x_{v+1}^2) = \bigwedge_{(i,j)}(x_{ij}^2)$.

Se $x_{kr}^2 < l_{kr}$, então determine um caminho p de r para k de maior habilidade com respeito a x_{v+1}^2 e vá ao passo 4.

Se $l_{kr} \leq x_{kr}^2 \leq u_{kr}$, então substitua \bar{x}_{v+1}^2 e vá ao passo 5.

Se $x_{kr}^2 > u_{kr}$, então determine um caminho p do nó k ao r de maior habilidade com respeito a x_{v+1}^2 e vá ao passo 4.

Passo 4: se $\mu_{kr}(x_{kr}^2) \geq \mu_C^p(x_{v+1}^2)$, então substitua $\bar{x}_{v+1}^2 \leftarrow x_{v+1}^2$ e vá ao passo 5. Caso contrário, transforme o fluxo x_{v+1}^2 no ciclo formado pelo arco (k, r) e o caminho como a seguir:

$$x_{ij}^2 \begin{cases} x_{ij}^2 + 1 & \text{se } (i, j) \in \vec{p} \\ x_{ij}^2 - 1 & \text{se } (i, j) \in \overleftarrow{p} \end{cases} \quad (8.8)$$

$$x_{kr}^2 \begin{cases} x_{kr}^2 + 1 & \text{se } x_{kr}^2 < l_{kr} \\ x_{kr}^2 - 1 & \text{se } x_{kr}^2 > u_{kr} \end{cases} \quad (8.9)$$

e vá ao passo 3.

Passo 5: confira se a condição $\mu_D(\bar{x}_v^1) \geq \mu_D(x_{v+1}^2)$ é válida. Caso seja, então o fluxo \bar{x}_v^1 é ótimo, FIM. Caso contrário, substitua $\bar{x}^1 \leftarrow \bar{x}_{v+1}^2$ e vá ao passo 2.

8.2.5 Grafo crisp e capacidade do arco fuzzy: fluxo contínuo

Este problema foi desenvolvido em (Chanas e Kolodziejczyk, 1984). O fluxo e as capacidades dos arcos são números fuzzy.

Um intervalo de fluxos admissíveis no arco $(i, j) \in A$ não é dado pelo intervalo $[b_{ij}, c_{ij}]$, mas na forma de um intervalo fuzzy, denotado por C_{ij} . O valor $\mu_{ij}(x_{ij})$ para um dado arco x_{ij} pode ser considerado tanto como o grau de satisfação da restrição de capacidade fuzzy pelo fluxo no arco x_{ij} como a possibilidade de realização do fluxo x_{ij} no arco (i, j) .

O objetivo a ser alcançado será: a partir de um fluxo determinado pelo usuário v , encontrar o grau de satisfação para aquele valor S . Em resumo, o problema pode ser reduzido a uma decisão fuzzy, utilizando as equações 8.4, 8.5 e 8.6

Definição 8.2.3. Um conjunto fuzzy V sobre \mathfrak{R} com a função de pertinência $\mu_V(v) = \bigvee_{x_v} \mu_C(x_v)$ é chamada uma capacidade fuzzy da rede.

Definição 8.2.4. A capacidade fuzzy do corte (X, \bar{X}) é o número fuzzy $C(X, \bar{X})$ definido como:

(i) para uma rede como restrição de capacidade inferior e superior:

$$C(X, \bar{X}) = \sum_{(i,j) \in \overrightarrow{(X, \bar{X})}} C_{ij} - \sum_{(i,j) \in \overleftarrow{(X, \bar{X})}} C_{ij}$$

(ii) para um rede com capacidade superior

$$C(X, \bar{X}) = \sum_{(i,j) \in \overrightarrow{(X, \bar{X})}} C_{ij}$$

Teorema 8.2.3. Seja W o conjunto de todos os cortes na rede e I o conjunto fuzzy tal que $\mu_I(x) = \bigvee_v \mu_V(v)$, $\forall x \in \mathfrak{R}$. Então a seguinte relação vale:

$$V = I \bigcap \left(\bigcap_{(X, \bar{X}) \in W} C(X, \bar{X}) \right)$$

As propriedades estabelecidas dão base para o seguinte algoritmo:

Algoritmo (Chanas e Kolodziejczyk, 1984):

Passo 1: Determine o fluxo real máximo x_w e o respectivo corte mínimo (X, \bar{X}) na rede com capacidade u_{ij} . Se $\mu_R(w) = 1$, então x_w é ótimo. Caso contrário, vá ao passo 2.

Passo 2: Determine o número fuzzy $C(X, \bar{X})$ e coordenadas (v, p) do ponto de intersecção das funções $\mu_{C(X, \bar{X})}$ e μ_R . Se $p = 0$, então o problema não é possível, FIM. Caso contrário, vá ao passo 3.

Passo 3: Determine o fluxo máximo x_w e o respectivo corte mínimo (X, \bar{X}) em G com capacidades u_{ij}^p . Se $\mu_D(x_w) = p$ e $w = v$, então o fluxo x_w é ótimo. Caso contrário, vá ao passo 2.

8.2.6 Proposta de algoritmo para o problema de fluxo máximo com capacidade do arco fuzzy

Vamos considerar que o decisor não possui conhecimento prévio da rede e que não pode opinar sobre o fluxo aceitável para o problema. Um algoritmo será desenvolvido tal que μ_G não toma parte no algoritmo. Todo conjunto de soluções será apresentado ao decisor. Portanto, não encontraremos um α -nível com a solução ótima, mas a solução ótima para cada α -nível.

Seja o seguinte exemplo (Malik e Moderson, 2001):

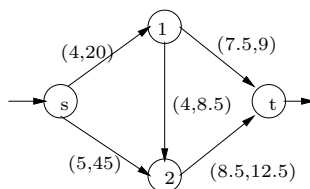


Figura 8.2: Exemplo de um problema do fluxo máximo com capacidade fuzzy com (c_{ij}, u_{ij})

Os cortes essenciais deste exemplo são $[X, \bar{X}]_1 = (\{s\}, \{1, 2, t\})$, $[X, \bar{X}]_2 = (\{s, 2\}, \{1, t\})$, $[X, \bar{X}]_3 = (\{s, 1, 2\}, \{t\})$ e $[X, \bar{X}]_4 = (\{s, 1\}, \{2, t\})$ com $C[X, \bar{X}]_1 = (9, 65)$, $C[X, \bar{X}]_2 = (12.5, 32.5)$, $C[X, \bar{X}]_3 = (16, 21.5)$ e $C[X, \bar{X}]_4 = (16.5, 62.5)$.

Observe o gráfico da Figura 8.3 referente aos valores dos cortes essenciais:

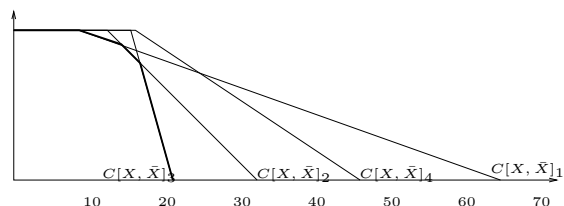


Figura 8.3: Valores dos cortes essenciais do exemplo

O algoritmo de (Chanas e Kolodziejczyk, 1984) analisa os valores dos três primeiros cortes,

que são aqueles que determinam o conjunto solução do problema. Como podemos notar pelo gráfico da Figura 8.3, o corte (1) é o corte mínimo para valores de fluxos que variam de $[0,14.4)$ com pertinência entre 1 e 0.9. O corte essencial (2) determina o corte mínimo no intervalo $[14.4,17.33)$ e possui graus de pertinência 0.9 a 0.76. O corte (3) determina o corte mínimo entre $[17.33,21.5]$ com pertinência variando de 0.76 a 0.

Chanas e Kolodziejczyk (1984) afirmam que o número máximo de cortes essenciais em uma rede é igual a 2^{m-2} . Um algoritmo deve ser feito de modo a tomar apenas aqueles que fazem parte da solução. Intuitivamente, o conjunto de solução será construído adicionando-se os cortes essenciais ao conjunto; fazendo-se testes nas intersecções dos valores dos cortes, até que todos os cortes significativos estejam contidos no conjunto solução S .

Sejam os seguintes parâmetros:

$\tilde{c}_{ij} = (c_{ij}, u_{ij})$: custo fuzzy tal que

$$\mu_{ij}(u) = \begin{cases} 1, & \text{se } u \leq c_{ij} \\ R(u; \tilde{c}_{ij}), & \text{se } c_{ij} \leq u \leq u_{ij} \\ 0, & \text{se } u > u_{ij} \end{cases}$$

onde $R(u; \tilde{c}_{ij})$ uma função monotônica decrescente.

$C[X, \bar{X}]$: soma das capacidades fuzzy dos arcos que pertencem ao corte essencial $[X, \bar{X}]$.

P : conjunto com as intersecções dos valores dos cortes essenciais.

S : conjunto com todos os cortes essenciais que possuem grau de pertinência não nulo para algum valor de fluxo v .

Vamos apresentar agora o algoritmo proposto para o problema:

Algoritmo proposto para o problema de fluxo máximo com capacidade fuzzy: FF-MAX

Passo 0: Seja $\tilde{c}_{ij} = (c_{ij}, u_{ij})$, $S = \emptyset$ e $P = \emptyset$. Resolva o problema G_1 para c_{ij} ($\alpha = 1$) e G_0 para u_{ij} ($\alpha = 0$). Vá para o passo 1.

Passo 1: Se $[X, \bar{X}]_1 = [X, \bar{X}]_0$, então

$$S \leftarrow [X, \bar{X}]_1$$

e encerre o algoritmo. Caso contrário, encontre o ponto de intersecção r do valores de corte $C[X, \bar{X}]_1$ e $C[X, \bar{X}]_0$. Faça $S \leftarrow S \cup \{[X, \bar{X}]_0, [X, \bar{X}]_1\}$, $P \leftarrow P \cup \{(r, v)\}$ e vá ao passo 2.

Passo 2: Enquanto $P \neq \emptyset$, execute o passo 3. Caso contrário, encontre os intervalos para cada corte mínimo e encerre o programa.

Passo 3: Faça $P \leftarrow P - \{(r, v)\}$. Sejam os valores crisp de capacidade $(\tilde{c}_{ij})_r$ para cada arco da rede. Encontre o corte mínimo $[X, \bar{X}]_r$ para este problema. Se $C[X, \bar{X}]_r = v$, volte ao passo 2. Caso contrário, faça:

$$S \leftarrow S \cup \{[X, \bar{X}]_r\}$$

encontre os pontos de intersecção s de $C[X, \bar{X}]_r$ comparando com todos os cortes essenciais pertencentes a S . Para novo s encontrado, faça:

$$P \leftarrow P \cup \{(s, v)\}$$

e volte ao passo 2.

O algoritmo proposto encontra todos os cortes que participam da solução do problema. Os intervalos dados pelo grau de pertinência em que cada conjunto essencial predomina é calculada da seguinte forma:

$$(\underline{r}, \bar{r}) = (\max_{\beta_i < r} \{\beta_i\}, \min_{\beta_i > r} \{\beta_i\})$$

com β_i sendo os pontos de intersecção do valor de corte $C[X, \bar{X}]_r$ com os outros valores pertencentes a S .

Contudo, este conjunto pode ter um tamanho indesejável ao decisor. A complexidade do algoritmo é dependente do número de conjuntos de corte da rede (até 2^{m-2} cortes). Nos casos mais complexos não podemos garantir a eficiência do algoritmo. Neste caso, apresenta-se aqui um algoritmo aproximado semelhante aos algoritmos propostos para o caso de estrutura fuzzy (Delgado et al., 1990).

Além dos parâmetros apresentados anteriormente, os seguintes parâmetros são necessários:

$\bar{\alpha}$: valor de pertinência mínimo que o fluxo pode assumir.

H : número de partições do intervalo $[\bar{\alpha}, 1]$

Heurística para o problema de fluxo máximo: FFMAX-H

Passo 1: Seja $S = \emptyset$. Seja também $\bar{\alpha}$ e H . Para $\alpha = \bar{\alpha} + i * h$, $h = \frac{1-\bar{\alpha}}{H-1}$, $i = 0, \dots, H - 1$ execute o passo 2.

Passo 2: Seja $(\tilde{c}_{ij})_\alpha$ o valor crisp da capacidade do arco (i, j) com pertinência α . Resolva o problema de fluxo máximo crisp. Faça:

$$S \leftarrow S \cup \{[X, \bar{X}]_\alpha\}$$

e retorne ao passo 1.

Passo 3 Devolva S .

Então, para cada α -nível têm-se um conjunto de corte essencial. O número de conjuntos $[X, \bar{X}]$ será, no máximo, igual ao número de partições do problema. Portanto, H e o algoritmo utilizado para resolver o problema em cada nível determinam a complexidade deste procedimento.

8.2.7 Exemplo

No exemplo da Figura 8.2 temos uma rede com parâmetros fuzzy que gera quatro conjuntos de corte, como descrito na Seção 8.2.6. Vamos considerar o algoritmo proposto por (Chanas e Kolodziejczyk, 1984), o algoritmo proposto neste trabalho e o algoritmo aproximado.

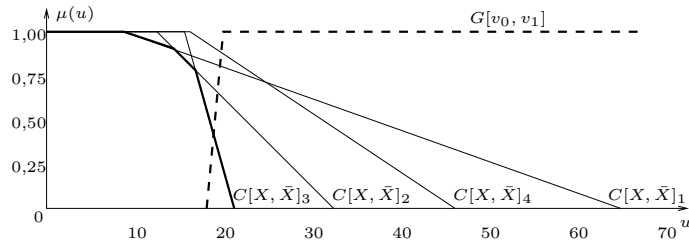


Figura 8.4: Cortes essenciais, região ótima e valor de satisfação proposto por Chanas (1984)

Por Malik e Moderson (2001), os passos do algoritmo de (Chanas e Kolodziejczyk, 1984) são:

Passo 1: Fluxo máximo real $v = 9$ e corte mínimo $[X, \bar{X}] = (\{s\}, \{1, 2, t\})$. $\mu_D(9) = 0$. Então vá ao passo 2.

Passo 2: $C[X, \bar{X}] = (9, 65)$. Ponto de intersecção $(v, r) = (19.62, 0.81)$. Como $r = 0.81 \neq 0$, vá ao passo 3.

Passo 3: $c_{s1}^r = 7.04$, $c_{s2}^r = 12.6$, $c_{12}^r = 4.86$, $c_{1t}^r = 7.78$, $c_{2t}^r = 9.26$. Fluxo máximo para c_{ij}^r , $w = 16.3$. $[X, \bar{X}] = (\{s, 2\}, \{1, t\})$. $\mu_D(16.3) = 0 \neq 0.81$ e $w = 16.3 \neq 19.62 = v$, então vá ao passo 2.

Passo 2: $C[X, \bar{X}] = (12.5, 32.5)$, $(v, r) = (19.32, 0.66)$ e $r = 0.66 \neq 0$, então vá ao passo 3.

Passo 3: $c_{s1}^r = 9.44$, $c_{s2}^r = 13.6$, $c_{12}^r = 5.53$, $c_{1t}^r = 8.01$, $c_{2t}^r = 9.86$. Fluxo máximo para c_{ij}^r , $w = 17.87$. $[X, \bar{X}] = (\{s, 1, 2\}, \{t\})$. $\mu_D(17.87) = 0 \neq 0.66$ e $w = 17.87 \neq 19.32 = v$, então vá ao passo 2.

Passo 2: $C[X, \bar{X}] = (16, 21.5)$, $(v, r) = (18.93, 0.467)$ e $r = 0.467 \neq 0$, então vá ao passo 3.

Passo 3: $c_{s1}^r = 12.54$, $c_{s2}^r = 26.32$, $c_{12}^r = 8.3$, $c_{1t}^r = 10.63$, $c_{2t}^r = 18.93$. Fluxo máximo para c_{ij}^r , $w = 18.93$. $[X, \bar{X}] = (\{s, 1, 2\}, \{t\})$. $\mu_D(18.93) = 0.467 = r$ e $w = 18.93 = v$. FIM. O fluxo é ótimo.

Os passos do algoritmo proposto neste trabalho são:

Passo 0: $S = \emptyset$, $P = \emptyset$. Solução para o problema G_1 : $w = 9$ e $[X, \bar{X}]_1 = (\{s\}, \{1, 2, t\})$.

Solução para o problema G_0 : $w = 21.5$ e $[X, \bar{X}]_0 = (\{s, 1, 2\}, \{t\})$. Vá para o passo 1.

Passo 1: Como $[X, \bar{X}]_0 \neq [X, \bar{X}]_1$, então a intersecção entre $C[X, \bar{X}]_0 = (16, 21.5)$ e $C[X, \bar{X}]_1 = (9, 65)$ é igual a $r = 0.8614$ com $v = 16.7623$. $S \leftarrow S \cup \{[X, \bar{X}]_0, [X, \bar{X}]_1\}$, $P \leftarrow P \cup \{(r, v)\}$ e vá ao passo 2.

Passo 2: Como $P = \{(r, v)\}$, vá ao passo 3.

Passo 3: $P \leftarrow P - \{(r, v)\}$, com $r = 0.8614$. Sejam $c_{s1}^r = 6.2176$, $c_{s2}^r = 10.5440$, $c_{12}^r = 4.6237$, $c_{1t}^r = 7.7079$ e $c_{2t}^r = 9.0544$, então $w = 15.2720$ e $[X, \bar{X}]_{0.8614} = (\{s, 2\}, \{1, t\})$ e $C[X, \bar{X}]_{0.8614} = (12.5, 32.5)$. Como $w|v=16.7623$, então os pontos de intersecção de $[X, \bar{X}]_{0.8614}$ com os elementos de S são $r_0 = 0.7586$ com $v_0 = 17.3280$ e $r_1 = 0.9028$ com $v_1 = 14.4432$. Então $S = \{[X, \bar{S}]_0, [X, \bar{X}]_1, [X, \bar{X}]_{0.8614}\}$ e $P = \{(r_0, v_0), (r_1, v_1)\}$. Volte ao passo 2.

Passo 2: Como $P \neq \emptyset$, vá ao passo 3.

Passo 3: $P \leftarrow P - \{(r_0, v_0)\}$. A solução da rede com $c_{ij}^{0.7586}$ é igual a $w = 17.3280$. Como $w = v_0$, então volte ao passo 2.

Passo 2: Como $P \neq \emptyset$, vá ao passo 3.

Passo 3: $P \leftarrow P - \{(r_1, v_1)\}$. A solução da rede com $c_{ij}^{0.9028}$ é igual a $w = 14.4432$. Como $w = v_0$, então volte ao passo 2.

Passo 2: Como $P = \emptyset$, $[X, \bar{X}]_0$ é o corte mínimo no intervalo $[9, 14.4432)$ com graus de pertinência entre $[1, 0.9028)$, $[X, \bar{X}]_{0.8614}$ é o corte mínimo no intervalo $[14.4432, 17.3280)$ com graus de pertinência ente $[0.9028, 0.7586)$ e o corte $[X, \bar{X}]_0$ é o corte mínimo no intervalo $[17.3280, 21.5)$ com graus de pertinência entre $[0.7586, 0]$. Fim do algoritmo.

Para o algoritmo aproximado, com um número de partições do intervalo, $H = 6$, que encontre os três cortes com $\bar{\alpha} = 0$, temos então

Passo 1: Seja $S = \emptyset$, $\bar{\alpha} = 0$ e $H = 6$. Para $\alpha = 0$ ($i=0$), vá ao passo 2.

Passo 2: Seja $(\tilde{c}_{ij})_0 = u_{ij}$. A solução do problema é $[X, \bar{X}]_0 = (\{s, 1, 2\}, \{t\})$ e $v = 21.5$. Então $S \leftarrow (\{s, 1, 2\}, \{t\})$. Volte ao passo 1.

Passo 2 com $\alpha = 0.2$ Seja $(\tilde{c}_{ij})_{0.2} = (16.8, 37, 7.6, 8.7, 11.7)$, a solução do problema é $[X, \bar{X}]_{0.2} = (\{s, 1, 2\}, \{t\})$ e $v = 20.4$. Então $S \leftarrow S \cup \{(\{s, 1, 2\}, \{t\})\}$. Volte ao passo 1.

Passo 2 com $\alpha = 0.4$ Seja $(\tilde{c}_{ij})_{0.2} = (13.6, 29, 6.7, 8.4, 10.9)$, a solução do problema é $[X, \bar{X}]_{0.4} = (\{s, 1, 2\}, \{t\})$ e $v = 19.3$. Então $S \leftarrow S \cup \{(\{s, 1, 2\}, \{t\})\}$. Volte ao passo 1.

Passo 2 com $\alpha = 0.6$ Seja $(\tilde{c}_{ij})_{0.2} = (10.4, 21, 5.8, 8.10, 10.1)$, a solução do problema é $[X, \bar{X}]_{0.6} = (\{s, 1, 2\}, \{t\})$ e $v = 18.2$. Então $S \leftarrow S \cup \{(\{s, 1, 2\}, \{t\})\}$. Volte ao passo 1.

Passo 2 com $\alpha = 0.8$ Seja $(\tilde{c}_{ij})_{0.2} = (7.2, 13, 4.9, 7.8, 9.3)$, a solução do problema é $[X, \bar{X}]_{0.4} = (\{s, 2\}, \{1, t\})$ e $v = 16.5$. Então $S \leftarrow S \cup \{(\{s, 2\}, \{1, t\})\}$. Volte ao passo 1.

Passo 2 com $\alpha = 1.0$ Seja $(\tilde{c}_{ij})_{0.2} = (4, 5, 4, 7.5, 8.5)$, a solução do problema é $[X, \bar{X}]_1 = (\{s\}, \{1, 2, t\})$ e $v =$. Então $S \leftarrow S \cup \{(\{s\}, \{1, 2, t\})\}$.

Passo 3: Devolva $S = \{(\{s, 1, 2\}, \{t\}), (\{s, 2\}, \{1, t\}), (\{s\}, \{1, 2, t\})\}$.

O algoritmo de Chanas e Kolodziejczyk (1984) encontra a melhor solução para a informação fornecida pelo decisor e devolve o fluxo no nível encontrado. O algoritmo proposto neste trabalho encontra os cortes que formam a região da solução e a devolve ao decisor, que pode utilizar uma informação mais completa para suas avaliações da rede. O algoritmo aproximado pode ser usado para casos mais complexos. No exemplo dado acima ele não se mostrou eficiente, pois foi necessário um número de partições, H , elevado em relação ao número de cortes que realmente participam da solução para encontrar o politopo completo. Contudo, se o decisor precisar de uma informação em um problema de grande porte, pode-se primeiro aplicar o algoritmo aproximado para definir uma região aproximada e estabelecer um objetivo mais factível com o espaço de solução e aplicar o algoritmo de (Chanas e Kolodziejczyk, 1984).

8.2.8 Caso II: Estrutura do grafo fuzzy

O caso estudado para um problema com estrutura fuzzy foi o problema de transporte fuzzy (Chanas et al., 1995). Neste problema ele faz definições sobre fluxo fuzzy, sobre o problema do fluxo máximo fuzzy em uma rede de transporte fuzzy. A seguir, apresentaremos as informações relevantes deste trabalho para a resolução de um problema de fluxo máximo com a estrutura do grafo fuzzy.

8.2.9 Definição de fluxo fuzzy

Para definir fluxo em um grafo fuzzy será necessária a teoria desenvolvida em (Delgado et al., 1985).

Definição 8.2.5. *Define-se nível de conectividade, $C(G)$, de um grafo fuzzy $G = (\sigma, \mu)$ como:*

$$C(G) = \min\{\mu_{ij}^\infty | i, j \in N; i \neq j\}$$

G é conectado se, e somente se, $C(G) > 0$. É fácil ver que G^α é conectado, no sentido crisp, $\forall \alpha \in (0, C(G)]$. Isto é, $C(G)$ é o limitante superior dos níveis de pertinências correspondentes aos α -cortes .

Definição 8.2.6. *A função ciclomática é uma aplicação $h_G(\cdot)$ em $(0, 1]$ para o conjunto de inteiros não-negativos tais que $\forall \alpha \in (0, 1]$, h_G é o número ciclomático de G^α . Isto é, denota-se por n^α , m^α e p^α o número de nós, arcos e componentes conectados (no sentido fraco) de G^α , então*

$$h_G(\alpha) = m^\alpha - n^\alpha + p^\alpha$$

(Chanas et al., 1995) definem algumas propriedades para as definições acima e chega a seguinte definição de fluxo em um grafo fuzzy:

Definição 8.2.7. *Considerando o grafo fuzzy G com $C(G) > 0$ e n arcos com funções de pertinência não nulos, qualquer vetor $\mathbb{R}^n \phi$ é um fluxo em G se, e somente se, $\exists \alpha \in (0, C(G)]$ tal que ϕ é um fluxo em G^α .*

Se ϕ é um fluxo em G^α , então é um fluxo para qualquer G^β , $\beta \leq \alpha$. Portanto, ϕ é um fluxo no nível α se este é o máximo valor de pertinência para o qual ϕ é dito ser um fluxo. Denota-se ϕ/α .

Com isso, podemos definir o conjunto fuzzy de fluxos $\Phi \in G$ como

$$\Phi = \bigcup_{\alpha} \phi/\alpha$$

8.2.10 O problema do fluxo máximo em um grafo fuzzy

Considerando as definições sobre fluxo fuzzy da Seção 8.2.9 anterior, pode-se definir o problema do fluxo máximo em um grafo fuzzy.

Considerando uma rede de transporte fuzzy $G = (\sigma, \mu)$ com capacidade associada $k_v(\cdot)$, definimos o fluxo máximo para G como o seguinte subconjunto fuzzy de A , Ω :

$$\forall \phi/A(\phi) > 0, \Omega(\phi) = \begin{cases} A(\phi), & \text{se } \forall \xi/A(\xi) \geq A(\phi) \Rightarrow \phi_0 \geq \xi_0 \\ 0, & \text{caso contrário.} \end{cases} \quad (8.10)$$

o problema é, então encontrar o conjunto fuzzy Ω .

Propriedade 8.2.1. *Seja $\xi_\alpha \forall \alpha \in (0, 1]$, o conjunto de fluxo obtido pela resolução do problema do fluxo máximo em G^α com capacidade $k_v(\alpha)$. O valor do fluxo máximo é dado por $\Xi = \bigcup_{\alpha} \xi_\alpha$.*

Com base na propriedade acima e utilizando os algoritmos para grafos fuzzy como nos Capítulos anteriores, podemos resolver este problema. A definição 8.2.6 define os intervalos em que serão calculados os níveis de solução do problema.

Algoritmo (Chanas et al., 1995):

Para $l = 1, \dots, s$ faça

 Calcule o fluxo ótimo em G^{η_l} usando

$$\forall v \in U^{\eta_l}, k_v(\beta) \in (\eta_{l-1}, \eta_l)$$

 como função de capacidade.

A aplicação deste algoritmo e o exemplo podem ser encontrados em (Chanas et al., 1995).

Capítulo 9

Problema de fluxo de custo mínimo fuzzy

Este é o último problema estudado neste trabalho. É um dos principais problemas em fluxos em redes. Aqui a incerteza pode ocorrer em mais de um parâmetro. Para estudar o problema crisp temos as seguintes referências: (Ahuja et al., 1993), (Bazaraa et al., 1990) e (Goldbarg e Luna, 2000).

9.1 Introdução

O problema de fluxo de custo mínimo (PFCM) consiste em atender, com custo mínimo, a demanda em uma rede, dada a oferta de recursos e a restrição de capacidade dos arcos. Os problemas de caminho mínimo (Capítulo 5) e fluxo máximo (Capítulo 8) são casos especiais do PFCM.

Seja uma rede $G = (N, A)$ com custo c_{ij} associado a cada unidade de fluxo, x_{ij} , que passa pelo arco $(i, j) \in A$. Para cada arco (i, j) , temos uma capacidade mínima l_{ij} e máxima u_{ij} . Para cada nó temos um parâmetro b_i , sendo que se:

$b_i > 0$, então o nó é um centro produtor. Exemplo: uma fábrica.

$b_i = 0$, então este é um nó de transbordo. Exemplo: um armazém.

$b_i < 0$, então o nó é de consumo. Exemplo: clientes.

Considera-se que $\sum_{i \in N} b_i = 0$, isto é à quantidade total de demanda é igual a quantidade de

matéria-prima disponível nos nós fonte. Este problema pode ser modelado da seguinte forma:

$$\text{Min} \quad z = \sum_{(i,j) \in A} c_{ij} x_{ij} \quad (9.1)$$

$$\text{s.a.} \quad \sum_{(k,i) \in A} x_{ki} - \sum_{(i,k) \in A} x_{ik} = b_i, \quad \forall i \in N \quad (9.2)$$

$$l_{ij} \leq x_{ij} \leq u_{ij}, \quad (i, j) \in A \quad (9.3)$$

sendo que a Equação (9.1) é a função objetivo do problema (minimizar o custo total do fluxo na rede), a restrição (9.2) é referente a conservação de fluxo em cada nó e (9.3) é a restrição de capacidade dos arcos. Nesta última restrição vamos considerar, sem perda de generalidade, que $l_{ij} = 0$.

Existem vários algoritmos desenvolvidos para a resolução do PFCM. Alguns mais conhecidos são o método simplex para redes e o algoritmo Out-of-Kilter, que estão descritos em (Bazaraa et al., 1990). Existem outros que podem ser estudados em (Ahuja et al., 1993), sendo alguns deles algoritmos polinomiais.

Para o caso fuzzy, foi encontrado apenas um trabalho para o PFCM. (Shih e Lee, 1999) fazem uma adaptação do método Húngaro para encontrar uma solução para o problema de fluxo de custo mínimo fuzzy, utilizando programação possibilística, para problemas com múltiplos níveis. Neste caso, o problema resultante a ser resolvido é crisp.

9.1.1 Considerações sobre a incerteza no problema de fluxo de custo mínimo

Considerando a estrutura do problema crisp, temos que a matriz de incidência nó-arco, A , é crisp. Para satisfazer as restrições de conservação de fluxo nos nós, a variável x_{ij} também é considerada crisp. Os demais parâmetros a seguir podem ser tratados como valores incertos:

c_{ij} : o custo pode ser representado por um número fuzzy. É o estudo mais freqüente para problemas em grafos com parâmetros fuzzy.

u_{ij} : pode-se permitir uma tolerância na capacidade dos arcos, associada a uma função de pertinência. Um estudo com capacidades fuzzy dos arcos já foi realizada para o problema de fluxo máximo (Chanas, 1987).

b_i : a oferta e demanda nos nós pode não ser precisa. Este tipo de estudo ainda não foi encontrado na literatura e não vai ser tratado neste trabalho. A incerteza nestes valores implicaria em rever a restrição de conservação de fluxo da rede pois o fluxo pode ser um número fuzzy. Ainda não é possível fazer este tipo de estudo baseados nos trabalhos atuais.

Neste capítulo, os dois parâmetros a serem estudados, c_{ij} e u_{ij} , podem apresentar incerteza nos valores. Isto nos leva a observar a relação entre eles. Uma solução deste problema que atende a demanda na rede sem violar a capacidade dos arcos (isto é, grau de pertinência do fluxo igual a 1) pode apresentar um alto custo. Por outro lado, havendo uma violação na capacidade podemos ter uma solução factível com grau de pertinência igual a α , $\alpha \in [0, 1]$, e um custo menor.

Primeiro, um estudo considerando separadamente cada variável foi realizado com intuito de observar o comportamento individual. A partir destas observações, um algoritmo geral é proposto.

9.2 Problema de fluxo de custo mínimo com incerteza na capacidade dos arcos

Considere o problema clássico apresentado na Seção 9.1, mas com a capacidade dos arcos desta rede representadas por números fuzzy \tilde{u}_{ij} como na Figura 9.1:

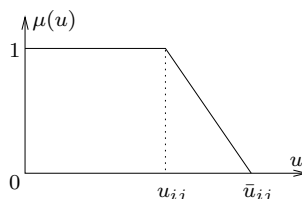


Figura 9.1: Capacidade fuzzy de um arco

Nesta figura, podemos observar que o uso de um arco por um fluxo (crisp) tem associado um grau de pertinência $\mu_{ij}(u)$, $u \in [0, \bar{u}_{ij}]$ e $\mu_{ij}(u) \in [0, 1]$. No caso de $u \leq u_{ij}$ o valor de pertinência é igual a 1 e pode ser considerado como no caso crisp.

Para obter uma solução *factível*, segundo algum grau de pertinência, primeiro temos que proceder um estudo sobre a incerteza em relação à capacidade nos arcos. Neste caso, o tratamento não é o mesmo fornecido pelo problema de fluxo máximo, pois o objetivo aqui é de atender a uma demanda conhecida na rede e não de descobrir qual o maior fluxo que a rede suporta. Portanto, a observação dos conjuntos de corte não se aplica aqui.

No entanto, se procedermos α -cortes podemos encontrar soluções com graus de pertinência iguais a α . Isto é, o problema aceita violar a capacidade de determinados arcos, mas a solução tem grau de pertinência α , ou um grau de *confiabilidade*, no conjunto de soluções ótimas. Para cada α -corte, resolve-se um problema crisp por meio de algoritmos conhecidos. Com isso teremos

um conjunto de soluções, sendo que a medida em que se reduz o custo, também se reduz o grau de confiabilidade no atendimento do fluxo na solução. Este raciocínio segue a do trabalho de (Delgado et al., 1990) para problemas com grau de pertinência associados aos nós e arcos de um grafo e pode ser visto na Seção 4.2.

Isto nos leva a um algoritmo simples para construção do conjunto de soluções, similar ao caso do problema com estrutura de grafos fuzzy.

9.2.1 Algoritmo proposto para o problema de fluxo de custo mínimo com capacidade fuzzy: PFCM01

Passo 0: Seja um grafo $G = (N, A)$ com custo c_{ij} , $(i, j) \in A$ e capacidade fuzzy \tilde{u}_{ij} . Defina o número de partições do intervalo h . Para $\alpha_k = \frac{k}{h}$, $k = 1, \dots, h$, execute o passo 1:

Passo 1: Resolva o problema crisp associado com $u_{ij} = (\tilde{u}_{ij})_\alpha$. Seja F a solução encontrada para o problema de fluxo de custo mínimo. Faça:

$$\begin{aligned} F_k &\leftarrow F & (9.4) \\ C_k &\leftarrow \sum_{(i,j) \in F} c_{ij} \\ \alpha_k &\leftarrow \bigwedge_{(i,j) \in F} \mu_{ij} = \min_{(i,j) \in F} (\mu_{ij}) \end{aligned}$$

onde α_k é o valor de pertinência da solução F_k com custo C_k . Volte ao passo 0.

9.3 Problema de fluxo de custo mínimo com custo dos arcos fuzzy

Neste caso, o custo associado a cada arco é um número fuzzy. Este problema não é tão fácil de se analisar quanto os outros, pois não temos conjunto de corte como no problema de fluxo máximo (veja Capítulo 8) e a análise do valor de possibilidade do problema é diferente do observado por (Okada, 2004), em que a interatividade dos arcos das soluções factíveis é analisada.

Neste caso, teremos um fluxo associado a cada arco, que pode não ser igual para cada solução a ser comparada. Isso torna mais complexo o cálculo do grau de possibilidade para cada arco, pois este valor pode ser dependente do fluxo.

No caso de considerarmos todo o conjunto de arcos por onde passa o fluxo solução (factível) do problema e não cada parte deste conjunto, seria uma comparação de números fuzzy simples, como realizada utilizando o valor de possibilidade dado pela função de (Dubois e Prade, 1980), descrita na equação 2.6 e procedendo como nos capítulos anteriores. Mas como tratar a interatividade das soluções neste caso? É possível fazer isso?

Vamos observar um exemplo da Figura 9.2:

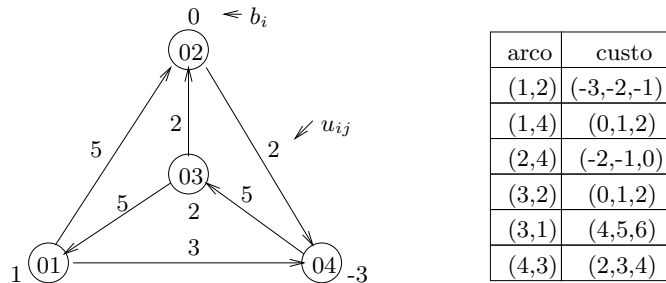


Figura 9.2: Exemplo de um problema de fluxo de custo mínimo

com u_{ij} a capacidade de cada arco, b_i a capacidade de cada nó e \tilde{c}_{ij} o comprimento fuzzy dado na tabela ao lado do exemplo, sendo que o comprimento é um número triangular fuzzy. Sejam algumas soluções factíveis encontradas, apresentadas na Figura 9.3:

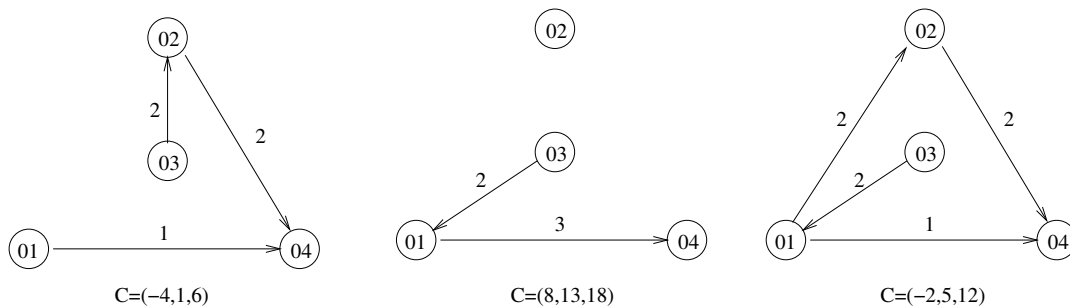


Figura 9.3: Soluções do problema anterior: (a), (b), (c)

Podemos notar pelos custos do fluxo resultante, C , que existe um valor de possibilidade de cada um ser a melhor solução. Mas os fluxos nos arcos são diversos.

Neste trabalho, a princípio vamos considerar o cálculo do valor de possibilidade do modo não interativo. Com isso, nosso problema resume-se a encontrar as soluções factíveis, visto que os algoritmos desenvolvidos para o problema de fluxo de custo mínimo encontram uma solução factível no final da execução.

Para obter soluções factíveis, vamos trabalhar com as capacidades dos arcos para desviar fluxo para rotas alternativas. Com isso, pretendemos enumerar todas as soluções possíveis. A

partir de uma solução crisp baseada em valores modais (ver Capítulo 4), vamos tentar obter outras soluções factíveis para construção do conjunto solução fuzzy. Em cada arco desta solução crisp, tenta-se desviar o fluxo para outras sub-rotas. Quando isto não for possível, proíbe-se este arco de sofrer outras alterações e tomamos outro arco da solução para repetir o procedimento até que não exista arco para tentar desviar o fluxo.

9.3.1 Algoritmo

Seja um grafo $G = (N, A)$ com custo \tilde{c}_{ij} , $(i, j) \in A$ e capacidade u_{ij} . Seja α dado.

Passo 0: Resolva o problema crisp associado sendo $c_{ij} = \min_{\mu_{c_{ij}}(u)=1} u$. Seja F_{\max} a solução do problema. Faça $S \leftarrow \{\frac{F_{\max}}{1}\}$ e $\bar{F} \leftarrow F_{\max}$

Passo 1: Verifique quais arcos de \bar{F} , não proibidos, possuem $x_{ij} = u_{ij}$ e dentre eles, qual possui a menor capacidade. Tente desviar uma unidade de fluxo para outro caminho. Calcule o valor de possibilidade entre esta nova solução F com F_{\max} . Se o valor de possibilidade, β , for maior do que α , então faça $S \leftarrow S \cup \{\frac{F}{\beta}\}$ e $u_{ij} \leftarrow u_{ij} - 1$ e $\bar{F} \leftarrow F$. Caso contrário, proíba este arco de sofrer alteração em sua capacidade e vá para o passo 2:

Passo 2: Atribua o valor original de capacidade aos arcos proibidos. Se ainda houver arcos que podem sofrer alterações em sua capacidade, volte ao Passo 1. Caso contrário, FIM.

9.4 Confiabilidade vs. Otimalidade do conjunto

Nos casos anteriores, cada parâmetro foi estudado separadamente. Quando observa-se a incerteza na capacidade, verifica-se que o conjunto solução encontrado possui soluções em que o fluxo nos arcos têm confiabilidade dada pelo valor de pertinência no conjunto. Quando estudamos o custo fuzzy dos arcos, verificamos que o fluxo assume um valor de possibilidade de ser a solução com menor custo fuzzy.

Quando estamos em um problema em que tanto a capacidade quanto o custo são números fuzzy, então no Passo 1 do algoritmo proposto na Seção 9.2.1, ao invés de resolver o problema crisp, resolve-se o problema fuzzy. Neste caso, como seria a relação entre a confiabilidade da solução e a otimalidade? No caso, pode-se utilizar o valor de pertinência em relação à capacidade dos arcos como valor de corte para o conjunto solução. Assim, para um valor de confiabilidade

mais alto, o conjunto de soluções seria mais reduzido também. Outras abordagens podem ser estudadas.

9.4.1 Algoritmo proposto para o problema de fluxo de custo mínimo com custo fuzzy: PFCM02

Passo 0: Seja um grafo $G = (N, A)$ com custo \tilde{c}_{ij} , $(i, j) \in A$ e capacidade \tilde{u}_{ij} . Defina o número de partições do intervalo h . Para $\alpha_k = \frac{k}{h}$, $k = 1, \dots, h$, execute o passo 1:

Passo 1: Resolva o problema com custos fuzzy, capacidade igual a $(u_{ij})_\alpha$, e valor de corte do conjunto solução igual a α utilizando o algoritmo da Seção 9.3.1. Seja F a solução encontrada para o problema de fluxo de custo mínimo. Faça:

$$\begin{aligned} \tilde{F}_k &\leftarrow F \\ \tilde{C}_k &\leftarrow \sum_{(i,j) \in \tilde{F}} \tilde{c}_{ij} \\ \alpha_k &\leftarrow \bigwedge_{(i,j) \in \tilde{F}} \mu_{ij} = \min_{(i,j) \in \tilde{F}} (\mu_{ij}) \end{aligned} \tag{9.5}$$

onde α_k é o valor de confiabilidade da solução \tilde{F}_k com custo \tilde{C}_k . Faça $k \leftarrow k + 1$

9.4.2 Exemplo

Vamos tomar o exemplo da Figura 9.2. Se calcularmos o problema crisp, com custo dos arcos igual ao valor modal (vide Capítulo 4), a primeira solução da Figura 9.3(a) é obtida como solução de maior grau de possibilidade.

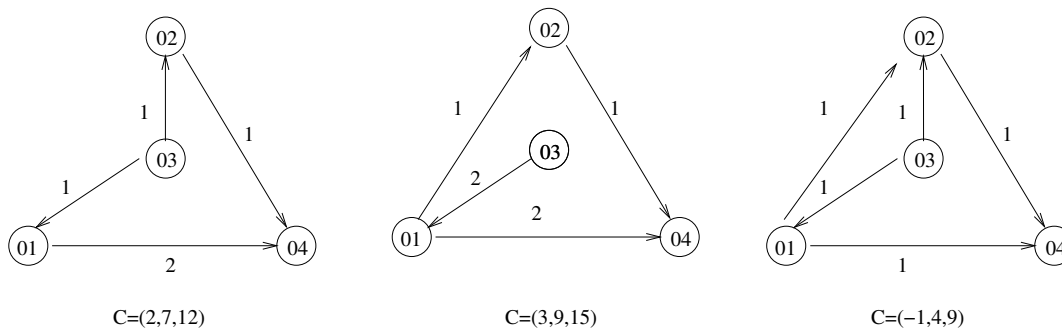


Figura 9.4: Outras soluções do exemplo: (d),(e) e (f)

Neste exemplo, ao tentarmos retirar o arco (1, 4), percebemos que não existe solução factível que não use este arco. Então, passamos a observar outro arco (3, 2). Retirando uma unidade de fluxo deste arco e desviando, obtemos outra solução, dada na Figura 9.4(d) e (e). Desviando duas unidades de fluxo obtemos a terceira solução da Figura 9.3 (c). Passando para o arco (2, 4); retirando uma unidade de fluxo obtemos a Figura 9.4(f) e se retirarmos as duas unidades de fluxos, obtém-se a segunda solução da Figura 9.3(b). Neste exemplo, não existem outras soluções. Em problemas maiores, deve-se observar soluções que não utilizam os arcos da solução de maior grau de possibilidade para verificar o grau destas soluções.

O grau de possibilidade das soluções apresentadas serem a solução de custo mínimo é dada na tabela 9.1

	(a)	(b)	(c)	(d)	(e)	(f)
<i>D</i>	1,0000	0,0000	0,6667	0,4000	0,2727	0,7000

Tabela 9.1: Grau de possibilidade de cada solução

Neste tipo de problema, calcular o grau de pertinência de cada arco pertencente a uma solução requer cuidados, pois a estes arcos estão associados fluxos. Este tipo de abordagem será tratada em trabalhos futuros. Uma das opções possíveis é a determinação do grau de pertinência do fluxo em um arco, ao invés de calcular o grau de pertinência do arco na solução.

O problema de fluxo de custo mínimo ainda está em fase inicial de estudo, visto a dificuldade em se tratar os diferentes parâmetros associados. Os algoritmos aqui sugeridos são apenas viáveis para problemas pequenos, mas sugerem algumas questões que podem ser estudadas.

Capítulo 10

Conclusões e trabalhos futuros

Neste trabalho, estudou-se alguns tipos de incerteza nos problemas clássicos de grafos e fluxos em redes, do ponto de vista dos algoritmos e da teoria associada. Foi realizado um levantamento bibliográfico e os trabalhos relevantes foram estudados e comentados no decorrer deste trabalho. Os problemas abordados: caminho mínimo (Capítulo 5), árvore geradora mínima (Capítulo 6), emparelhamento (Capítulo 7), fluxo máximo (Capítulo 8) e fluxo de custo mínimo (Capítulo 9), foram discutidos e os seus algoritmos foram apresentados, melhorados e/ou desenvolvidos.

Para o problema do caminho mínimo fuzzy, Capítulo 5, foi feita uma revisão bibliográfica sobre os trabalhos desenvolvidos e os algoritmos mais relevantes foram descritos. Uma modificação do algoritmo de Okada (2001) para o problema de caminho mínimo com custo fuzzy foi apresentada, visando a redução no número de comparações de soluções e o descarte de caminhos que não participam do conjunto solução. Para o problema de caminho mínimo em que a estrutura de grafos é fuzzy, foi apresentada uma adaptação do algoritmo fundamental proposto por (Delgado et al., 1990) para o problema. Além disso, heurísticas para encontrar subconjuntos de soluções e uma heurística para encontrar o caminho com maior grau de pertinência foram propostos.

No Capítulo 6, o problema da árvore geradora mínima foi estudado. O algoritmo para o problema com parâmetros fuzzy foi desenvolvido, considerando o trabalho de (Okada, 2001) para construir o conjunto solução do problema. Para este problema, também foi proposto um algoritmo genético para encontrar um conjunto solução aproximado. Para o problema com estrutura do grafo fuzzy, foi apresentado o trabalho de (Delgado et al., 1990) e (Chunde, 1996). Algumas heurísticas foram propostas, dentre elas a que encontra a árvore geradora com maior grau de pertinência.

Em relação aos problemas de emparelhamento, Capítulo 7, os algoritmos exato e genético

para o problema de designação foram desenvolvidos neste trabalho, baseados em observações feitas para o problema da árvore geradora mínima. O problema do casamento estável foi estudado por (Nair, 2000) e apresentado neste trabalho como um dos problemas que podem ser abordados em emparelhamento.

Chanas (1987) desenvolveu uma série de trabalhos referente ao problema de fluxo máximo, em que incertezas são associadas tanto à capacidade quanto à estrutura do grafo. No Capítulo 8, são apresentados os algoritmos mais relevantes da literatura. Um algoritmo exato e uma heurística foram desenvolvidos, considerando que não existe a informação do grau de satisfação do decisor. Para o problema em que a estrutura do grafo é fuzzy, o trabalho de (Chanas et al., 1995) é descrito.

Finalmente, para o problema de fluxo de custo mínimo foi desenvolvido, no Capítulo 9, um algoritmo para encontrar o conjunto solução do problema, considerando custo e/ou capacidade fuzzy. Dada a dificuldade em se tratar um problema em que dois parâmetros são fuzzy, apenas problemas pequenos podem ser tratados pelo algoritmo descrito.

A incerteza pode estar em diversos níveis, a solução requisitada pelo decisor pode ser diversa, dependendo das suas necessidades. Os algoritmos exatos apresentados podem ser aprimorados através de melhorias na programação e no estudo da comparação dos candidatos à solução.

No entanto, as heurísticas e meta-heurísticas abordadas em cada Capítulo mostraram-se atraentes. Considerando o tipo de solução requerida e o tamanho do problema, as heurísticas apresentam uma solução aproximada em um tempo computacional significativamente reduzido. Um estudo sobre a diversidade da população inicial nos algoritmos genéticos deve ser realizado em trabalhos futuros para garantir uma exploração mais eficiente do espaço de solução. Também pode ser feito um estudo para outras técnicas heurísticas ou meta-heurísticas, notadamente os algoritmos meméticos para obter as melhores soluções do conjunto solução fuzzy (Buriol, 2000) e das técnicas de sistemas imunológicos artificiais (Silva, 2001). Por exemplo, seleção clonal, tendo em vista que o operador de mutação foi o mais eficiente nos algoritmos genéticos implementados.

Outros problemas também podem ser estudados sob a teoria de conjuntos fuzzy:

Problemas de clustering: com trabalhos iniciais feitos por (Yeh e Bang, 1975), mas existem algoritmos eficientes para este problema considerando redes neurais fuzzy ao invés de utilizar sua estrutura de grafos.

problemas de transporte: veja exemplo em (Chanas et al., 1995).

problemas de multicommodity, dentre outros.

estes serão objeto de trabalhos futuros, com o desenvolvimento prévio da teoria necessária para viabilizar o estudo destes problemas.

Referências Bibliográficas

- Ahuja, R., Magnati, T. e Orlin, J. (1993). *Network flows : theory, algorithms, and applications*, Prentice-Hall.
- Alaoui, A. (1999). On fuzzification of some concepts of graphs, *Fuzzy Sets and Systems* **101**: 363–389.
- Ali, M., Ramamurthy, B. e Deogun, J. (2000). Routing and wavelength assignment with power considerations in optical networks, *Computer Networks* **32**: 539–555.
- Bazaraa, M., Jarvis, J. e H.F.Sherali (1990). *Linear programming and network flows*, second edn, John Wiley.
- Bellman, R. e Zadeh, L. (1970). Decision-making in a fuzzy environment, *Management Science* **17**(4): B-141–B-164.
- Bershtein, L. e Dziouba, T. A. (2000). Allocation of maximal bipartite part from fuzzy graph, *ESIT2000*, pp. 207–211.
- Berthold, M. e Huber, K.-P. (1995). Neural network based construction of fuzzy graphs, *Proceedings of the 2nd Annual Conference on Information Science*.
- Bhattacharya, P. (1987). Some remarks on fuzzy graphs, *Pattern Recognition Letters* **6**: 297–302.
- Bhattacharya, P. e Suraweera, F. (1991). An algorithm to compute the supremum of max-min powers and a property of fuzzy graphs, *Pattern Recognition Letters* **12**: 413–420.
- Bhutani, K. R. (1989). On automorphism of fuzzy graphs, *Pattern Recognition Letters* **9**: 159–162.
- Bhutani, K. R. e Battou, A. (2003). On m-strong fuzzy graphs, *Information Sciences* (xx): xx–xx.

- Blue, M., Bush, B. e Puckett, J. (1997). Applications of fuzzy logic to graph theory, *Technical Report LA-UR-96-4792*, Los Alamos National Laboratory.
- Blue, M. P., Bush, B. W. e Puckett, J. (2002). Unified approach to fuzzy graph problems, *Fuzzy Sets and Systems* **125**(3): 355–368.
- Buriol, L. S. (2000). *Algoritmos meméticos como parte de um framework para solução do problema do caixeiro viajante assimétrico*, Master's thesis, State University of Campinas.
- Chanas, S. (1987). *Fuzzy optimization in networks*, Dordrecht.
- Chanas, S., Delgado, M., Verdegay, J. e Vila, M. (1995). Fuzzy optimal flow on imprecise structures, *European Journal of Operational Research* **83**(3): 568–580.
- Chanas, S. e Kamburowski, J. (1981). The use of fuzzy variables in pert, *Fuzzy Sets and Systems* **5**: 11–19.
- Chanas, S. e Kasperski, A. (2003). Possible and necessary optimality of solutions in the single machine scheduling problem with fuzzy parameters, *Fuzzy Sets and Systems* (xx): xx–xx.
- Chanas, S. e Kolodziejczyk, W. (1982). Maximum flow in a network with fuzzy arc capacities, *Fuzzy Sets and Systems* **8**: 165–173.
- Chanas, S. e Kolodziejczyk, W. (1984). Real-valued flows in a network with fuzzy arc capacities, *Fuzzy Sets and Systems* **13**: 139–151.
- Chanas, S. e Kolodziejczyk, W. (1986). Integer flows in network with fuzzy capacity constraints, *Networks* **16**: 17–31.
- Chanas, S. e Kuchta, D. (1996). A concept of the optimal solution of the transportation problem with fuzzy cost coefficients, *Fuzzy Sets and Systems* **82**: 299–305.
- Chanas, S. e Kuchta, D. (1998). Fuzzy integer transportation problem, *Fuzzy Sets and Systems* **98**: 291–298.
- Chang, P. e Lee, E. (1999). Fuzzy decision networks and deconvolution, *Computers and Mathematics with Applications* **37**(11–12): 53–63.
- Chepoi, V. e Dumitrescu, D. (1999). Fuzzy clustering with structural constraints, *Fuzzy Sets and Systems* **105**: 91–97.
- Chongshu, L. (1991). The theorems of decomposition and representation for fuzzy graphs, *Fuzzy Sets and Systems* **42**: 237–243.

- Chunde, Y. (1996). On the optimization problem of spanning tree in fuzzy network, *The Journal of China Universities of Posts and Telecommunications* **03**(02): 22–25.
- Craine, W. (1994). Characterizations of fuzzy interval graphs, *Fuzzy Sets and Systems* **68**: 181–193.
- Delgado, M., Verdegay, J. e Vila, M. (1985). On fuzzy tree definition, *European Journal of Operational Research* **22**: 243–249.
- Delgado, M., Verdegay, J. e Vila, M. (1990). On valuation and optimization problems in fuzzy graphs: a general approach and some particular cases, *ORSA J. Comput.* pp. 74–83.
- Dubois, D. e Prade, H. (1980). *Fuzzy Sets and Systems*, Academic Press.
- Eppstein, D. (1994). Finding the k shortest paths, *IEEE Symposium on Foundations of Computer Science*, pp. 154–165.
*citeseer.nj.nec.com/eppstein97finding.html
- Goldbarg, M. e Luna, H. (2000). *Otimização Combinatória e programação linear: modelos e algoritmos*, Editora Campus.
- Gondran, M. e Minoux, M. (1986). *Graph and algorithms*, Wiley-interscience series in discrete mathematics, John Wiley & Sons.
- Gudwin, R. R. e Zuben, F. J. V. (2000). Notas de aula do curso ia770 - computação evolutiva.
*<http://www.dca.fee.unicamp.br/gudwin>
- Harary, F. (1972). *Graph Theory*, Addison-Wesley.
- Herrera, F., Lozano, M. e Verdegay, J. (1994). Applying genetic algorithms in fuzzy optimization problems, *Fuzzy Systems & A.I. Reports and Letters* **3**(1): 39–52.
- Holland, J. (1992). 2nd edn, The University of Michigan Press.
- Huber, K.-P., Berthold, M. e Szczerbicka, H. (1996). Fuzzy graph based metamodeling, *Proceedings of the conference on Winter simulation*.
- Inuiguchi, M. e Ramik, J. (2000). Possibilistic linear programming: a brief review of fuzzy mathematical programming and a comparison with stochastic programming in portfolio selection problem, *Fuzzy Sets and Systems* **111**(1): 3–28.
- J.M.Adamo (1980). Fuzzy decision trees, *Fuzzy Sets and Systems* **4**: 207–219.

- Julstrom, B. e G.R.Raidl (2002). Initialization is robust in evolutionary algorithms that encode spanning trees as sets of edges, *Proceedings of the 2002 ACM Symposium on Applied Computing*, ACM Press, pp. 547–552.
- Kóczy, L. (1992). Fuzzy graphs in the evaluation and optimization of networks, *Fuzzy Sets and Systems* **46**: 307–319.
- Kim, K. e Roush, F. (1982). Fuzzy flows on networks, *Fuzzy Sets and Systems* **8**: 35–38.
- Klein, C. (1991). Fuzzy shortest path, *Fuzzy Sets and Systems* **39**: 7–41.
- Kuwano, H. (2001). The indices of ranking fuzzy sets based on possibility theory, *The Second International Conference on Nonlinear Analysis and Convex Analysis, NACA2001*.
- Lasdon, L. (1970). *Optimization Theory for Large Systems*, MacMillan.
- Lin, C.-J. e Wen, U.-P. (2004). A labeling algorithm for the fuzzy assignment problem, *Fuzzy Sets and Systems, to appear* **142**: 373–391.
- Lin, K.-C. e Chern, M.-S. (1993). The fuzzy shortest path problem and its most vital arcs, *Fuzzy Sets and Systems* **58**: 343–353.
- Malik, D. e Moderson, J. N. (2001). *Fuzzy Discrete Structures*, Vol. 58 of *Studies in Fuzziness and Soft Computing*, Springer-Verlag.
- Mares, M. e Horak, J. (1983). Fuzzy quantities in networks, *Fuzzy Sets and Systems* **10**: 123–134.
- Michalewicz, Z. (1996). *Genetic Algorithms + Data Structures = Evolution Programs*, Springer.
- Moderson, J. (1993). Fuzzy line graphs, *Pattern Recognition Letters* **14**: 381–384.
- Moderson, J. e Nair, P. (1996a). Cycles and cocycles of fuzzy graphs, *Information and Computer Science* **90**: 39–49.
- Moderson, J. e Nair, P. (1996b). Successor and source of (fuzzy) finite state machines and (fuzzy) directed graphs, *Intelligent Systems* **95**: 113–124.
- Moderson, J. e Nair, P. (2000). *Fuzzy Graphs and Fuzzy Hypergraphs*, Studies in Fuzziness and Soft Computing, Physica-Verlag.
- Moderson, J. e Peng, C.-S. (1994). Operations on fuzzy graphs, *Information Sciences* **79**: 159–170.

- Mukherjee, N. P. e Bhattacharya, P. (1986). Fuzzy groups: Some group-theoretic analogs, *Intelligent Systems* **39**: 247–268.
- Nair, P. (2000). Fuzzy marriage problem, *NAFIPS 2000: The 19th International Meeting of the North American Fuzzy Information Processing Society*.
- Nair, P. e Cheng, S.-C. (2001). Cliques and fuzzy cliques in fuzzy graphs, *Joint 9th IFSA World Congress and 20th NAFIPS International Conference*, pp. 2277–2280.
- Okada, S. (1999). Critical path analysis in project network with fuzzy activity times, *Proc. of the Eighth International Fuzzy Systems Association World Congress*, Vol. 1, pp. 510–513.
- Okada, S. (2001). Interactions among paths in fuzzy shortest path problems, *9th International Fuzzy Systems Association World Congress*, pp. 41–46.
- Okada, S. (2004). Fuzzy shortest path problems incorporating interactivity among paths, *Fuzzy Sets and Systems* **142**(3): 335–357.
- Okada, S. e Gen, M. (1993). Order relation between intervals and its application to shortest path problem, *Computers and Industrial Engineering* **25**(1-4): 147–150.
- Okada, S. e Gen, M. (1994). Fuzzy shortest path problem, *Computers Ind. Engng.* **1-4**: 465–468.
- Okada, S. e Soper, T. (2000). A shortest path problem on a network with fuzzy arc lengths, *Fuzzy Sets and Systems* **109**: 129–140.
- Pedrycz, W. e Gomide, F. (1998). *An Introduction to Fuzzy Sets: Analysis and Design*, MIT Press.
- Peng, C.-S. e Juang, T. T.-Y. (1993). Fuzzy network flow, *Congressus Numerantium* **98**: 156–162.
- Perchant, A. e Bloch, I. (2002). Fuzzy morphism between graphs, *Fuzzy Sets and Systems* (128): 149–168.
- Raidl, G. R. e Julstrom, B. A. (2001). Edge-sets: An effective evolutionary coding of spanning trees, *Technical Report TR-186-1-01-01*, Institute of Computer Graphics and Algorithms, Vienna University of Technology. To appear in the IEEE Transactions on Evolutionary Computation.
- Rosenfeld, A. (1971). Fuzzy groups, *Journal of Mathematical Analysis and Applications* **35**: 512–517.

- Rosenfeld, A. (1975). Fuzzy graphs, *Fuzzy Sets and Their Applications*, Academic Press, pp. 77–95.
- Sakawa, M., Nishizaki, I. e Uemura, Y. (2001). Interactive fuzzy programming for two-level linear and linear fractional production and assignment problems: a case study, *European Journal of Operational Research* (135): 142–157.
- Shih, H.-S. e Lee, E. S. (1999). Fuzzy multi-level minimum cost flow problems, *Fuzzy Sets and Systems* (107): 159–176.
- Silva, L. N. C. (2001). *Engenharia Imunológica: desenvolvimento e aplicação de ferramentas computacionais inspiradas em sistemas imunológicos artificiais*, PhD thesis, Universidade Estadual de Campinas - UNICAMP.
- Sunitha, M. e Vijayakumar, A. (1999). A characterization of fuzzy trees, *Intelligent Systems* **113**: 293–300.
- Takeda, E. (1973). Connectivity in fuzzy graphs, *Technology Reports of the Osaka University* (1121): 343–352.
- Tan, S., Yu, Y. e Wang, P.-Z. (1998). Building fuzzy graphs from samples of nonlinear functions, *Fuzzy Sets and Systems* (93): 337–352.
- Uesu, H., Yamashita, H., Yanai, M. e Tomita, M. (2001). Sociometry analysis applying fuzzy node fuzzy graph.
- Wilson, R. e Watkins, J. (1990). *Graphs: An introductory approach: a first course in discrete mathematics*, John Wiley.
- Yager, R. (1986). Paths of least resistance in possibilistic production systems, *Fuzzy Sets and Systems* (19): 121–132.
- Yao, J.-S. e Lin, F.-T. (2003). Fuzzy shortest path network problems with uncertain edge weights, *Journal of Information Science and Engineering* **19**: 329–351.
- Yeh, R. e Bang, S. (1975). Fuzzy graphs, fuzzy relations and their applications to cluster analysis, *Fuzzy Sets and Their Applications*, Academic Press, NY, pp. 125–149.
- Zadeh, L. (1965). Fuzzy sets, *Information and Control* **8**: 338–353.
- Zadeh, L. (1978). Fuzzy sets as a basis for a theory of possibility, *Fuzzy Sets and Systems* **1**: 3–28.

Zadeh, L. (1994). Fuzzy logic, neural networks, and soft computing, *Comm. of the ACM* **37**(3): 77–84.

Zadeh, L. (1997). Toward a theory of fuzzy information granulation and its centrality in human reasoning and fuzzy logic, *Fuzzy Sets and Systems* **90**: 111–127.

Zadeh, L. (n.d.). Fuzzy logic and the calculi of fuzzy rules, fuzzy graphs, and fuzzy probabilities, *Computers and Mathematics with Applications* **37**: 35.

Apêndice A

Noções sobre a Teoria de Grafos

Neste apêndice, apresentamos os conceitos básicos da teoria de grafos necessário para a compreensão deste trabalho. Para um estudo mais completo, sugerimos o Capítulo 2 de (Ahuja et al., 1993) e os capítulos subseqüentes com os problemas relativos a grafos. Outras referências utilizadas neste trabalho para os conceitos da teoria de grafos e os seus algoritmos são: (Harary, 1972), (Bazaraa et al., 1990), (Wilson e Watkins, 1990) e (Goldbarg e Luna, 2000).

Para facilitar o uso, os tópicos estão organizados em ordem alfabética.

Arco: um par ordenado (i, j) sendo i o nó de origem do arco e j o nó de destino. Neste caso, se a_{ij} é um arco, então $a_{ij} \neq a_{ji}$ (Figura A.1).



Figura A.1: Exemplo de um arco (i, j)

Aresta: um par não ordenado $\{i, j\}$. Neste caso, se a_{ij} é uma aresta, temos necessariamente que $a_{ij} = a_{ji}$ (Figura A.2).



Figura A.2: Exemplo de uma aresta $\{i, j\}$

Árvore: um subgrafo conectado e sem ciclos.

Árvore geradora: Seja um grafo $G = (N, A)$. Uma árvore geradora é um subgrafo gerador de G que não possui ciclos (Figura A.3).

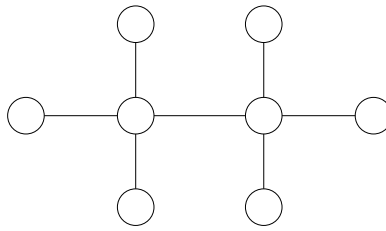


Figura A.3: Exemplo de uma árvore geradora

Árvore geradora mínima: Seja $G = (N, A)$ um grafo e $c_{ij} > 0$ os custos dos arcos (ou arestas). Uma árvore geradora mínima é uma árvore geradora, T , com menor custo total $C = \sum_{(i,j) \in T} c_{ij}$. Veja Figura 6.1, no Capítulo 6.

Cadeia: uma seqüência não orientada de nós e arestas, sem repetições de nós (Figura A.4).

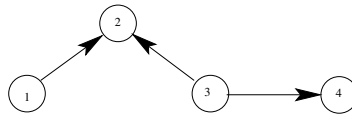


Figura A.4: Exemplo de uma cadeia

Caminho: uma seqüência orientada de nós e arcos, sem repetições de nós (Figura A.5).

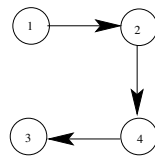


Figura A.5: Exemplo de um caminho

Capacidade de um arco (ou aresta): são valores referentes a quantidade mínima (l_{ij}) e máxima (u_{ij}) de fluxo (x_{ij}) que pode ser transportada pelo arco (ou aresta).

Ciclo: um cadeia fechada (Figura A.6).

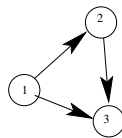


Figura A.6: Exemplo de um ciclo em um grafo

Circuito: uma caminho fechado (Figura A.7).

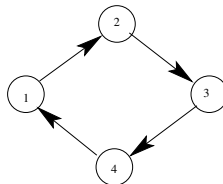


Figura A.7: Exemplo de um circuito em um grafo

Clique: um grafo onde existe um arco conectando cada par de nós.

Componente: é um subgrafo conectado de G que não é subgrafo próprio de nenhum outro subgrafo de G .

Conjunto de corte $[X, \bar{X}]$: Seja um grafo $G = (N, A)$. Seja $X \subset N$ e $\bar{X} = N - X$. $[X, \bar{X}] = \{(i, j) \in A | i \in X, j \in \bar{X}\}$, ou seja, um conjunto de corte é formado pelos arcos que conectam X a \bar{X} (Figura A.8).

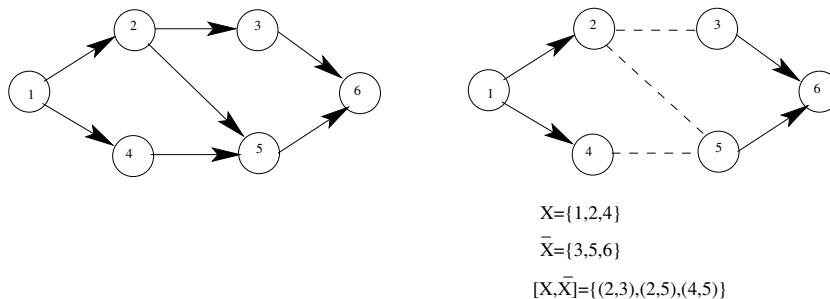


Figura A.8: Exemplo de um conjunto de corte $[X, \bar{X}]$

Custo de um arco: Para um problema de grafos é o valor associado ao escolher uma aresta (ou arco) para a solução de um problema. Se um fluxo estiver associado ao grafo, é o valor associado para se passar uma unidade de fluxo nesta aresta (ou arco), desde que não viole a capacidade deste.

Floresta: um grafo desconexo sendo que todas as suas componentes são árvores (Figura A.9).

Fluxo: é uma maneira de se enviar objetos, pessoas, etc. sobre arestas (ou arcos) de um nós para outro do grafo. Associamos números aos nós e arcos (ou arestas) de um grafo, normalmente designando uma capacidade (mínima e máxima) destes nós e arcos (ou arestas).

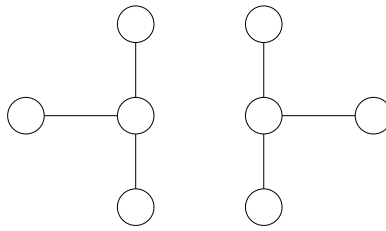


Figura A.9: Exemplo de floresta

Grafo: ou rede. Um par de conjuntos N e A , não vazios, finitos sendo que N é o conjunto de nós e A é o conjunto de arcos (ou arestas) tal que para $(i, j) \in A$ (ou $\{i, j\} \in A$), então $i, j \in N$.

OBS: Alguns autores não fazem distinção entre estes dois conceitos. Outros admitem que o conceito de rede pressupõe a presença de fluxo.

Grafo bipartido: Seja um grafo $G = (N, A)$. O grafo é bipartido se $N = N_1 \cup N_2$ tal que $N_1 \cap N_2 = \emptyset$, $|N_1| = |N_2| = n$ e para $(i, j) \in A$, temos que $i \in N_1$ e $j \in N_2$ (Figura A.10).

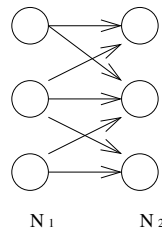


Figura A.10: Exemplo de um grafo bipartido

Grafo completo: um grafo onde existe um arco (ou aresta) conectada cada par de nós do grafo. Neste caso, temos para um grafo com n nós, $C_n^2 = \frac{n(n-1)}{2}$ arcos (ou arestas) (Figura A.11).

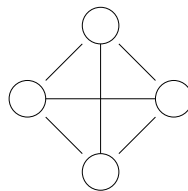


Figura A.11: Exemplo de grafo completo

Grafo conectado: um grafo (ou rede) é conectado se sempre existe um caminho (ou cadeia) ligando qualquer par de nós. Caso contrário o grafo é dito desconectado. Um grafo é dito

fortemente conectado se existe um caminho ligando dois nós quaisquer. Se existe uma cadeia ligando dois nós quaisquer, então o grafo é dito fracamente conectado.



Figura A.12: Exemplo de grafo fortemente e fracamente conectado

Grafo misto: um grafo $G = (N, A)$ onde A é um conjunto de arcos e arestas (Figura A.13).

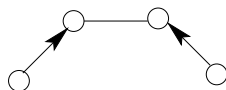


Figura A.13: Exemplo de um grafo misto

Grafo não orientado: um grafo $G = (N, A)$ onde A é um conjunto de arestas. Veja Figura A.11.

Grafo orientado (digrafo): um grafo $G = (N, A)$ onde A é um conjunto de arcos. Veja Figura A.14.

Grau de um nó: se refere ao número de arcos (ou arestas) incidentes no referido nó. Na Figura A.11, todos os nós tem o mesmo grau 3.

Laço: um arco (ou aresta) em que os nós de origem e destino do arco são os mesmo, ou seja (i, j) , $i = j$ (Figura A.14).

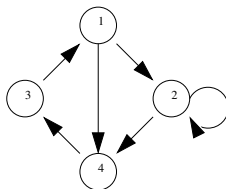


Figura A.14: Exemplo de um laço em um grafo orientado

Nó: também conhecidos como vértices ou pontos. Um dos conjuntos que caracterizam um grafo.

Nó-folha: ou nó terminal. Nó com grau 0 ou 1. Na Figura A.15, os nós s e t são nós terminais, ou nós folhas.

Nó-interno: qualquer nó que não seja um nó-folha.

Nó origem: Em um problema de fluxos em redes, um nó de origem é um nó capacitado. Se b_i é o parâmetro referente a capacidade do nó i , então $b_i > 0$. Está associado ao depósito, à matéria-prima, etc. Na Figura A.15, o nó s é um nó destino.

Nó destino: Em um problema de fluxos em redes, um nó de destino é um nó capacitado. Se b_i é o parâmetro referente a capacidade do nó i , então $b_i < 0$. Está associado ao consumidor, a demanda, etc. Na Figura A.15, o nó t é um nó destino.

Nó de transbordo: Em um problema de fluxos em redes, um nó de transbordo é um nó não capacitado. Se b_i é o parâmetro referente a capacidade do nó i , então $b_i = 0$. Na Figura A.15, os nós 1, 2, 3 e 4 são nós de transbordo (Figura A.15).

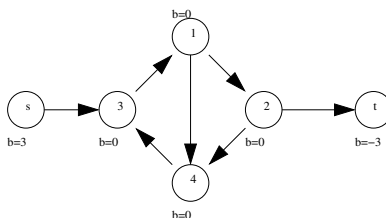


Figura A.15: Exemplo de nós capacitados

Rede: vide grafo.

Subgrafo: $G' = (N', A')$ do grafo $G = (N, A)$ é um subgrafo se $N' \subseteq N$ e $A' \subseteq A$.

Subgrafo gerador: é um subgrafo onde $N' \equiv N$ (Figura A.16).

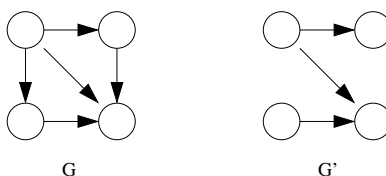


Figura A.16: Exemplo de um subgrafo gerador

Apêndice B

Noções sobre algoritmos genéticos

Neste apêndice, temos algumas noções de algoritmos genéticos. A intenção aqui não é a de produzir um texto completo sobre algoritmos genéticos, apenas disponibilizar a informação dos conceitos utilizados nesta dissertação, de modo a facilitar a compreensão dos algoritmos. Estas informações foram extraídas das notas de aulas em (Gudwin e Zuben, 2000). Para um estudo mais aprofundado, veja em (Michalewicz, 1996).

Desenvolvido inicialmente por Holland (Holland, 1992) nos anos 60, os algoritmos genéticos foram desenvolvidos para estudar formalmente os fenômenos de adaptação, naturais ou artificiais, com o propósito de importar estes mecanismos de adaptação para ambientes computacionais. Nesta época, os cientistas da computação já estudavam sistemas evolutivos com o objetivo e utilizá-los como uma ferramenta de otimização em problemas de engenharia (Gudwin e Zuben, 2000).

Um algoritmo genético é considerado um método fraco (um método genérico aplicado em um ambiente genérico) que deve ser utilizado apenas quando métodos fortes (métodos genéricos em ambientes específicos) e/ou específicos (métodos específicos em ambientes específicos) falham. Embora não garantam eficiência total na obtenção da solução, geralmente garantem uma boa aproximação. Outra característica é que os algoritmos genéticos operam em problemas não-lineares, não estacionários e problemas com explosão combinatória de candidatos a solução, uma grande vantagem em relação a outras técnicas (Gudwin e Zuben, 2000).

As regras de evolução são simples: as espécies evoluem aleatoriamente (via operadores de mutação, recombinação e outros), estando sujeitas à seleção natural sob recursos limitados. Os indivíduos mais adaptados sobrevivem e se reproduzem, propagando seu material genético às próximas gerações.

Para facilitar a leitura desta dissertação, os conceitos foram organizados em ordem alfabética, como em um pequeno glossário:

Alelos: dois genes que se encontram nas mesmas posições de dois cromossomos e são responsáveis por uma determinada característica.

Codificação: em algoritmos genéticos, consiste na determinação da estrutura do cromossomo. Os algoritmos genéticos clássicos adotam a codificação binária, isto é, cada gene (bit) pode assumir valores 0 ou 1. Outras codificações são a inteira e a real. Respectivamente, nestas codificações valores inteiros ou reais são associados a cada bit. No caso da codificação mista, mais de um tipo de codificação devem estar combinados em um cromossomo.

Cromossomo: estrutura nucleoprotéica formada por uma cadeia de DNA, sendo a base física dos genes nucleares, os quais estão dispostos linearmente. Em algoritmos genéticos, um cromossomo haplóide (apenas uma cadeia de DNA representa o cromossomo) geralmente corresponde a uma cadeia de bits que representa um candidato à solução do problema.

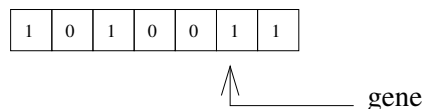


Figura B.1: Exemplo de cromossomo com codificação binária

Fenótipo: é a manifestação do genótipo no comportamento, fisiologia e morfologia do indivíduo, como um produto de sua interação com o ambiente.

Função de avaliação (fitness): medida que avalia a capacidade de adaptação e de reprodução de um indivíduo.

Genes: blocos funcionais de DNA, os quais codificam uma proteína específica. No caso do algoritmo genético, um gene corresponde a um único bit, ou então a um pequeno bloco de bits adjacentes que codificam um elemento particular da solução candidata.

Geração: Em algoritmos genéticos, uma iteração que consiste: seleção para reprodução, a reprodução e a seleção para a nova população.

Genoma: conjunto de todos os cromossomos que compõe o material genético do organismo.

Genótipo: representa o conjunto específico de genes do genoma. Nesta caso, indivíduos com o mesmo genoma são ditos terem o mesmo genótipo. Em algoritmos genéticos, uma solução

potencial de um dado problema pode ser codificada na forma de uma ou mais cadeias cromossômicas.

Mutação: um erro ocorrido durante a reprodução que causa a mudança de um determinado alelo por outro. Em algoritmos genéticos, uma perturbação na informação contida em um determinado gene. Dentre os operadores de mutação temos:

Mutação aleatória: Escolhe-se um indivíduo, sorteia-se a percentagem de genes que serão trocados e sorteia-se um valor para os genes, dentro do intervalo permitido pelo alelo.



Figura B.2: Exemplo de mutação simples

Mutação indutiva: Semelhante à mutação aleatória, só que o valor a ser sorteado para o alelo é somado ao valor atual do alelo, não substituído.



Figura B.3: Exemplo de mutação indutida em um cromossomo com codificação real

População: conjunto de indivíduos. Em algoritmos genéticos existem duas abordagens em relação a população. Na abordagem Michigan, a população como um todo é a solução do problema. Na abordagem Pittsburgh, cada indivíduo da população corresponde a uma solução do problema.

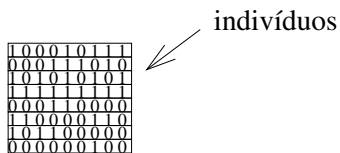


Figura B.4: Exemplo de uma população de oito indivíduos

Recombinação (crossover): consiste na troca aleatória de material genético entre dois cromossomos. Existem alguns tipos de recombinação, dentre eles:

Crossover simples (de um ponto): Escolhe-se dois indivíduos da população, determina-se o ponto de corte e efetua-se o crossover trocando as duas partes posteriores ao ponto de corte.

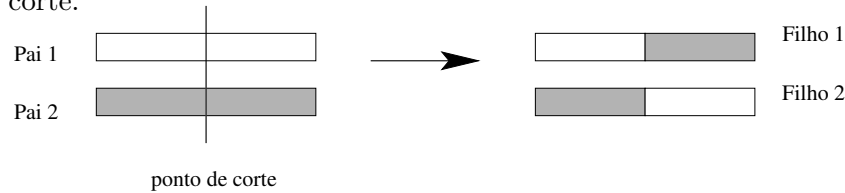


Figura B.5: Exemplo de um crossover de um ponto

Crossover de dois pontos: Escolhe-se dois indivíduos da população, determina-se dois pontos de corte e efetua-se o crossover trocando as partes do cromossomo entre os pontos de corte.

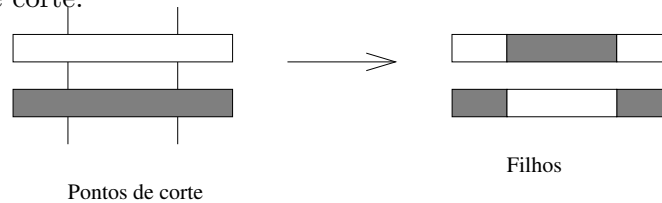


Figura B.6: Exemplo de um crossover de dois pontos

Crossover uniforme: Escolhe-se dois indivíduos da população e efetua-se o crossover trocando alguns alelos entre os dois indivíduos.

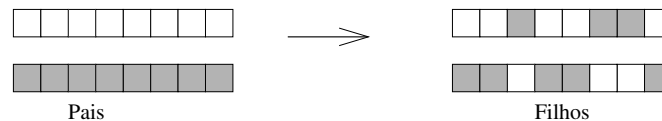


Figura B.7: Exemplo de um crossover de dois pontos

Crossover OX: Utilizado no problema do Caixeiro Viajante (Buriol, 2000), com a codificação inteira. Escolhe-se dois indivíduos da população, copia-se um trecho de um dos indivíduos e o restante do cromossomo do novo indivíduo é preenchido com as informações do outro indivíduo, na mesma sequência evitando valores repetidos nos genes.



Figura B.8: Exemplo de um crossover OX com codificação inteira

Reprodução: pode ser assexuada. Neste caso, a informação genética é transmitida sem alterações aos novos indivíduos, salvo mutações ocorridas durante a reprodução. No caso da reprodução sexuada, a informação de dois indivíduos pais são combinadas para a formação de um novo indivíduo.

Roulette Wheel (Roleta): uma técnica de seleção, em que a probabilidade de um cromossomo ser escolhido é diretamente proporcional ao seu valor na função de fitness.

Seleção: Geralmente a seleção é utilizada para escolher os indivíduos para a próxima geração. Os mecanismos mais comuns para a seleção são:

Elitista: são selecionados uma percentagem dos melhores indivíduos da população intermediária.

Aleatória salvacionista: o melhor indivíduo da população intermediária é selecionado para a nova geração e os indivíduos restantes são escolhidos aleatoriamente.

Aleatória não-salvacionista: a nova geração é escolhida aleatoriamente da população intermediária.

por Diversidade: são selecionados os indivíduos mais diversos na população intermediária, a partir do melhor indivíduo.

Bi-classista: São selecionados os $P\%$ melhores indivíduos e os $(100 - P)\%$ piores indivíduos.

por Torneio: dois indivíduos são escolhidos aleatoriamente. Um número aleatório r , entre 0 e 1 é gerado. Se $r < k$ (k um parâmetro dado) o melhor dos indivíduos é escolhido. Senão o outro é escolhido.

Steady-State: a população original é mantida, com a excessão de alguns indivíduos menos adaptados.

B.1 Um esboço do algoritmo genético utilizado neste trabalho

Aqui apresentamos um esquema básico do algoritmo genético utilizado nesta dissertação. Para cada problema, este algoritmo é adaptado: como gerar os indivíduos da população inicial e nos valores dos parâmetros associados; operadores de mutação e recombinação; a taxas de mutação e recombinação, percentagem de seleção de indivíduos para reprodução e para a nova geração, etc.

Passo 0: Criar a população inicial, com K indivíduos, segundo o critério adotado para cada problema. Seja P a população. Calcule o valor do fitness de cada indivíduo.

Passo 1: Para $I = 1$ até GER ou quando não houver mudanças em D em um determinado número de gerações, faça:

1. Selecione uma percentagem dos indivíduos de P para recombinação, proceda a operação e aloque os novos indivíduos na população intermediária (P_i).
2. Selecione uma percentagem dos indivíduos de P para a mutação, proceda a operação e aloque os indivíduos na P_i .
3. Selecione o melhor indivíduo da P e sorteie os restantes dos indivíduos que devem completar a P_i .
4. Faça $P \leftarrow P_i$. Calcule o valor do fitness de cada indivíduo de P .
5. Atualize D_{ij} (Equação 6.1).

Passo 2: Devolva D ao decisor.

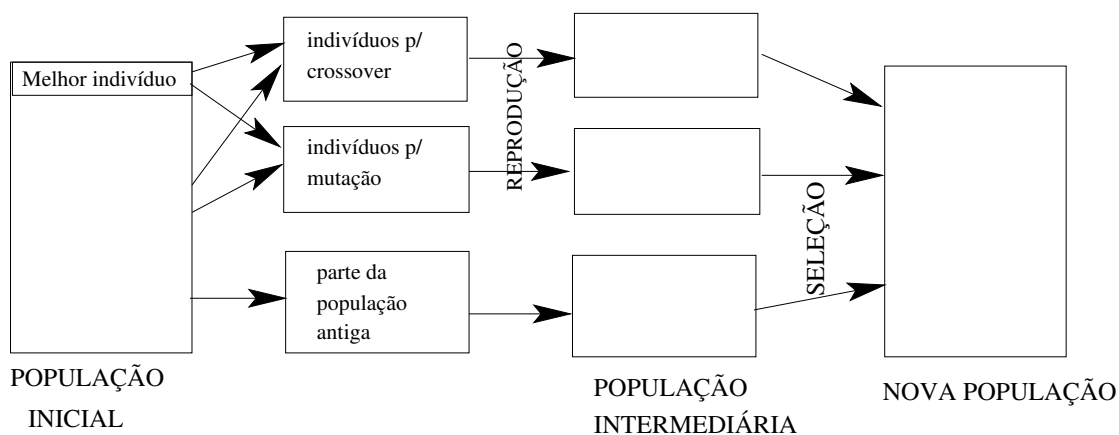


Figura B.9: Exemplo de um algoritmo genético

Apêndice C

Dados dos exemplos utilizados

C.1 Ali et al., 1999

Arcos	l	c	u	Arcos	l	c	u
01-Bolzano-Milano	126	140	154	19-Firenze-Roma	162	180	198
02-Bolzano-Verona	99	110	121	20-Pisa-Roma	171	190	209
03-Bolzano-Trieste	189	210	231	21-Ancona-Pescara	117	130	143
04-Milano-Verona	99	110	121	22-Perugia-Pescara	153	170	187
05-Milano-Torino	85	95	105	23-Perugia-Roma	108	120	132
06-Milano-Genova	81	90	99	24-Roma-Cagliari	410	460	506
07-Verona-Venezia	81	90	99	25-Roma-Napoli	162	180	198
08-Verona-Bolzano	85	95	105	26-Pescara-Napoli	180	200	198
09-Venezia-Trieste	76	85	94	27-Pescara-Bari	243	270	297
10-Venezia-Bolzano	85	95	105	28-Napoli-Bari	189	210	231
11-Torino-Genova	81	90	99	29-Napoli-Potenza	81	90	99
12-Genova-Bolzano	117	130	143	30-Napoli-Palermo	271	310	241
13-Genova-Firenze	135	150	165	31-Napoli-Catamin	315	350	385
14-Genova-Pisa	108	120	132	32-Bari-Potenza	90	100	110
15-Bolzano-Firenze	49	55	61	33-Cagliari-Palermo	378	420	462
16-Bolzano-Ancona	180	200	220	34-Potenza-Catanzaro	180	200	220
17-Firenze-Pisa	54	60	66	35-Catanzaro-Catamin	189	210	231
18-Firenze-Perugia	99	110	121	36-Palermo-Catamin	135	150	165

Tabela C.1: Distâncias entre as cidades da rede óptica italiana (em km)

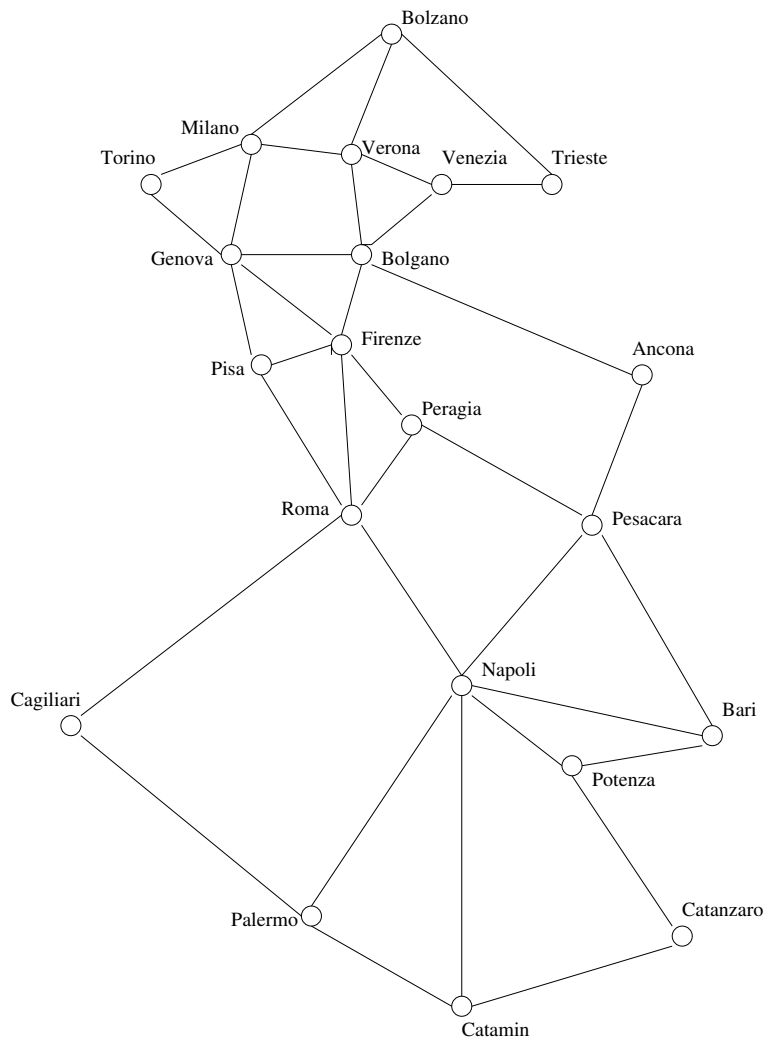


Figura C.1: Rede óptica italiana

C.2 Okada e Soper, 2000

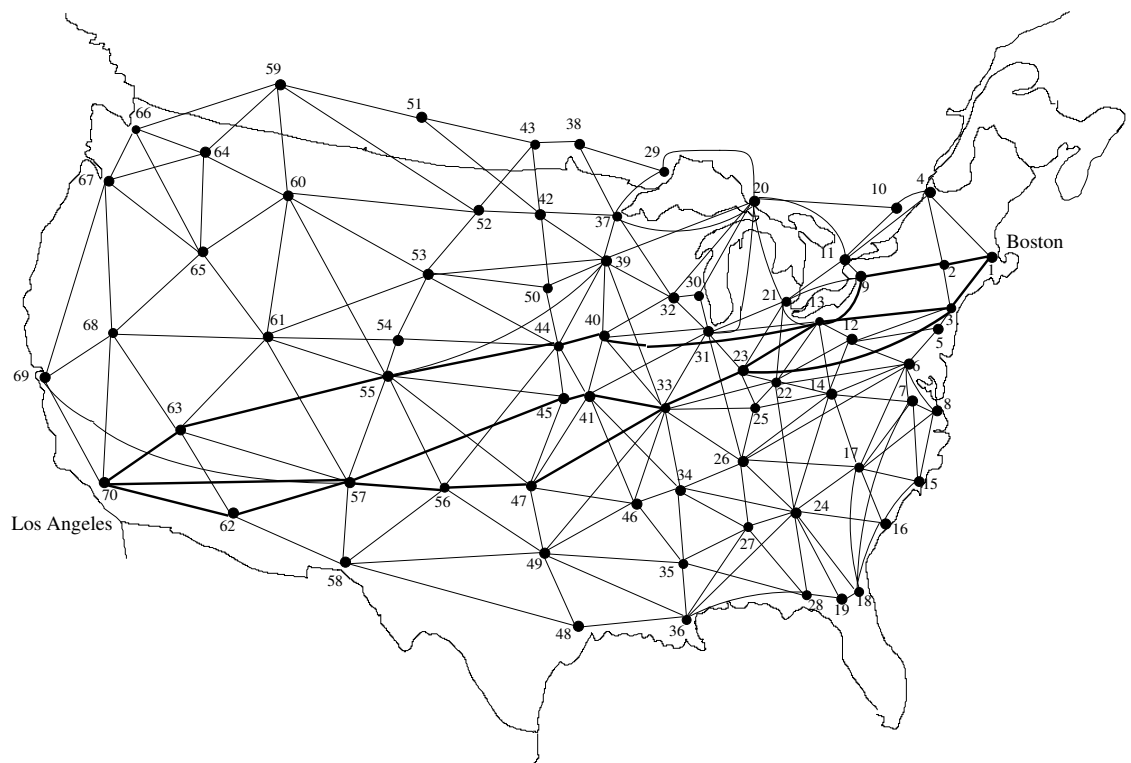


Figura C.2: Mapa dos Estados Unidos

n_o	n_d	t	n_o	n_d	t	n_o	n_d	t	n_o	n_d	t	n_o	n_d	t	n_o	n_d	t
1	2	206	11	21	275	21	23	328	32	40	346	42	52	232	56	60	953
1	3	259	12	13	154	21	31	335	33	34	348	43	51	429	56	61	596
1	4	349	12	14	272	22	23	137	33	39	664	43	52	490	56	63	902
2	3	186	12	22	350	22	25	120	33	40	438	44	45	194	57	58	320
2	4	220	13	14	332	22	24	577	33	41	306	44	50	208	57	61	744
2	9	354	13	21	196	22	33	418	33	46	427	44	53	695	57	62	556
3	5	109	13	22	295	23	25	133	33	47	398	44	54	594	57	63	692
3	9	456	13	23	366	23	31	216	33	49	385	44	56	646	57	69	1309
3	12	451	13	31	412	23	33	290	34	35	230	44	55	810	57	70	950
3	13	602	13	40	814	24	26	300	34	41	551	45	47	342	58	62	485
3	23	875	14	17	311	24	28	178	34	46	158	45	56	634	59	60	239
4	10	137	14	22	238	24	27	329	35	36	211	45	57	857	59	64	644
4	11	512	14	24	646	24	34	497	35	46	311	46	47	398	59	66	899
5	6	127	14	25	304	24	36	622	35	49	494	46	49	385	60	61	599
5	12	368	14	26	409	25	26	206	36	48	436	47	49	247	60	64	377
6	7	126	15	17	234	25	33	319	36	49	594	47	55	311	60	65	569
6	8	239	15	16	200	26	28	228	37	38	351	47	56	754	61	63	517
6	12	277	16	17	258	26	31	550	37	39	174	48	49	289	61	65	409
6	14	437	16	18	280	26	33	372	37	42	304	48	58	902	61	68	602
6	17	461	16	24	396	26	34	247	38	43	143	49	55	436	62	63	344
6	22	623	17	18	491	27	28	356	39	40	283	49	58	708	62	70	455
6	26	850	17	24	299	27	35	494	39	42	456	50	53	414	63	68	532
7	8	120	17	26	491	28	34	276	39	44	445	51	59	572	63	70	346
7	14	380	18	19	80	28	35	289	39	50	282	52	53	457	64	65	445
7	15	305	18	24	419	28	36	413	39	53	685	52	59	997	64	66	341
7	17	364	19	24	347	29	37	193	39	56	1045	52	60	832	64	67	419
7	18	778	19	27	124	29	38	485	40	41	232	53	54	345	65	66	592
8	15	293	20	21	420	30	31	114	40	44	162	53	60	730	65	67	520
8	17	406	20	29	528	30	32	96	41	44	216	53	61	884	65	68	503
9	11	58	20	30	514	31	32	172	41	45	91	54	56	114	66	67	208
9	13	229	20	31	589	31	33	355	41	46	448	54	61	516	67	68	799
9	21	317	20	32	540	31	40	408	41	47	442	55	56	508	67	69	766
10	11	297	20	37	510	31	41	606	42	43	188	55	57	346	68	69	263
10	20	600	20	39	725	32	37	402	42	50	288	55	58	498	68	70	601
11	20	547	21	22	328	32	39	324	42	51	613	56	57	536	69	70	452

Tabela C.2: Tempo (t , minutos) de percurso entre dois nós (cidades) n_o e n_d

C.3 Um grafo fuzzy completo com 10 nós

nó	1	2	3	4	5	6	7	8	9	10
1		0,6154	0,0579	0,0153	0,8381	0,1934	0,4966	0,7271	0,7948	0,1365
2			0,3529	0,7468	0,0196	0,6822	0,8998	0,3093	0,9568	0,0118
3				0,4451	0,6813	0,3028	0,8216	0,8385	0,5226	0,8939
4					0,3795	0,5417	0,6449	0,5681	0,8801	0,1991
5						0,1509	0,8180	0,3704	0,1730	0,2987
6							0,6602	0,7027	0,9797	0,6614
7								0,5466	0,2714	0,28844
8									0,2523	0,4692
9										0,0648
10										

Tabela C.3: Tabela com os valores de pertinência das arestas no grafo G

nó	1	2	3	4	5	6	7	8	9	10
1		2	6	10	7	6	5	10	9	8
2			7	6	7	9	6	9	8	8
3				10	9	7	10	8	8	6
4					7	9	5	6	6	8
5						5	6	5	9	6
6							10	5	6	8
7								6	9	8
8									8	8
9										8
10										

Tabela C.4: Tabela com os custos (pesos) das arestas no grafo G